

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SLEDOVÁNÍ OBJEKTŮ VE VIDEOSEKVENCÍCH

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JOZEF MLÍCH

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

SLEDOVÁNÍ OBJEKTŮ VE VIDEOSEKVENCÍCH

OBJECT TRACKING IN VIDEO SEQUENCES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. JOZEF MLÍCH

VEDOUCÍ PRÁCE
SUPERVISOR

Doc. Dr. Ing. PAVEL ZEMČÍK

BRNO 2008

Abstrakt

V této diplomové práci jsou představeny metody pro zpracování obrazu a metody, které statisticky modelují pohyb objektu. Z metod pro zpracování obrazu je blíže popsána metoda odečítání pozadí, která umožňuje detekci objektů v obraze. Dále jsou popsány morfologické operace jako je dilatace a eroze. Z metod pro statistické modelování pohybu objektu je popsán princip Kalmanova filtru a částicových filtrů.

Klíčová slova

počítačové vidění, Kalmanův filtr, Částicové filtry

Abstract

In this master thesis, image processing methods and methods for statistical modeling of motion are presented. First, description methods of image processing, such as background subtraction method used for object detection, are presented. Next, description of morphological operations, such as dilatation and erosion, is done. Finally, methods for statistical modeling, such as Kalman filter and particle filters, are shown.

Keywords

Computer vision, Kalman filter, particle filters

Citace

Jozef Mlích: Sledování objektů ve videosekvencích, diplomová práce, Brno, FIT VUT v Brně, 2008

Sledování objektů ve videosekvencích

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Doc. Dr. Ing. Pavla Zemčíka. Všechny literární prameny a publikace, ze kterých jsem čerpal, jsou řádně uvedeny.

.....
Jozef Mlích
26. května 2008

Poděkování

Velmi rád bych poděkoval vedoucímu mé diplomové práce Doc. Dr. Ing. Pavlu Zemčíkovi za odborné vedení a za to, že mi umožnil se podílet na výzkumu prováděném na ústavu počítačové grafiky. Dále bych chtěl poděkovat Ing. Víťovi Beranovi, Ing. Michalovi Hradišovi, Ing. Romanovi Juránkovi a Ing. Petrovi Chmelařovi za konzultace a odborné vedení. Část této diplomové práce týkající analýzy trajektorií byla financována z IST EU grantu Carretaker „Content Analysis and REtrieval Technologies to Apply Knowledge Extraction to massive Recording“ (FP6-027231). Některé data na kterých bylo provedeno vyhodnocení algoritmů pro sledování objektů byli poskytnuty organizací NIST v rámci projektu TrecVID.

© Jozef Mlích, 2008.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Získávání znalostí z videosekvencí	5
2.1 Přímé získávání znalostí	5
2.2 Blokový přístup	6
2.3 Hierarchický přístup	6
3 Zpracování obrazu	9
3.1 Barevné prostory	9
3.2 Filtace	11
3.3 Geometrické transformace	13
3.4 Extrakce příznaků	14
4 Počítačové vidění po snímcích	18
4.1 Segmentace obrazu	19
4.2 Shluková analýza	21
4.3 Vyhledání klasifikací	23
5 Počítačové vidění ve videosekvenci	25
5.1 Spárování objektu ve dvou snímcích	26
5.2 Problém překrytí	27
5.3 Analýza sekvenčních dat	32
6 Analýza chování a stavu objektů	34
6.1 Ověření správnosti sledování objektů	35
6.2 Kombinované události	36
7 Implementace	37
7.1 Integrace do programových systémů	37
7.2 Vyhledání objektů v obraze	38
7.3 Sledování objektů v obraze	39
7.4 Analýza chování objektů	42
8 Závěr	47
A Rozhraní modulu pro sledování objektů	52
B ER Diagram	55

Seznam obrázků

2.1	Schéma zpracování obrazu	8
3.1	Typický CRT gamut RGB	10
3.2	Barevné modely - zleva RGB, CMYK, HSV/HSL	10
3.3	3D Barevný histogram	16
3.4	Průmět barevného histogramu do 2D prostoru	16
3.5	Část hlavy člověka – obrázek použitý pro demonstraci tvorby histogramů	17
4.1	Maska popředí na základě dlouhodobého modelu pozadí	20
4.2	Maska popředí na základě krátkodobého modelu pozadí	21
4.3	Clustering metodou meanshift	22
4.4	Connected Component Labeling	23
5.1	Vliv rotace objektu na vzhled	25
5.2	Spárování objektů v sousedních snímcích	26
5.3	Extrakce trajektorie z grafu předpokládané polohy objektu	28
5.4	Normální rozložení pravděpodobnosti ve 2D	29
5.5	Schéma algoritmu Kalmanova filtru	30
5.6	Mechanismus převzorkování: Condensation algoritmus	31
5.7	Skrytý Markovův model	32
6.1	Detekce komplexního chování	36
7.1	Detekce objektu ve více snímcích	40
7.2	Vzdálenost nejbližšího objektu od anotace	41
7.3	Všechny normalní trajektorie pro scénu	43
7.4	ROC křivka – filtrované data	44
7.5	ROC křivka – nefiltrované trajektorie	45
7.6	Ukázka klasifikace trajektorií	46

Kapitola 1

Úvod

S rozvojem výpočetní techniky a video techniky dochází k vytváření a hromadění velkého množství záznamů obsahujících potencionálně zajímavá data. Ruční zpracování těchto audiovizuálních dat je neefektivní a nákladné, proto vzniká potřeba jejich automatického zpracování. Toto automatické zpracování zahrnuje úpravu těchto dat s cílem odstranit šum nebo jinak zlepšit vlastnosti obrazu a zvuku pro další zpracování nebo uložení těchto dat. Tyto metody lze použít například pro stabilizaci obrazu, úpravu jasu a kontrastu obrazu, kompresi dat.

Metody zpracování obrazu pro extrakci sémantických informací se nazývají počítačové vidění. Cílem těchto metod je přímý popis nějakého jevu v audiovizuálních datech na základě obrazových dat. Přesnost sémantických dat získaných pomocí počítačového vidění lze zlepšit analýzou sémantiky zvukových dat. Způsob uložení a interpretace sémantických dat pro potřeby proudového zpracování audiovizuálních dat je předmětem dalšího výzkumu.

Tato diplomová práce se zabývá aplikací metod pro zpracování obrazu, počítačové vidění, strojové učení, statistické modelování a simulace na video záznamy pořízené průmyslovými kamerami ve veřejné dopravě. Motivací této diplomové práce je řešení problémů spojených s projektem Caretaker [41].

Projekt Caretaker se zabývá analýzou obsahu video záznamů z veřejné dopravy. Cílem projektu je vyhodnocení možných bezpečnostních rizik a extrakce dalších potencionálně zajímavých informací. Mezi nejzávažnější bezpečnostní rizika patří hrozby teroristických útoku nebo požáru. Dalším typem bezpečnostních rizik, na které je kladen velký důraz, je kriminalita. Jako příklad lze uvést vandalismus, přepadení, střelbu nebo také pouhé přeskakování turniketů. Příkladem ostatních zajímavých informací, které lze získat z těchto videozáznamů jsou informace o případném zranění některého z cestujících, informace o vytíženosti nočních spojů nebo vytíženosti automatů na lístky. Jako asi nejobtížnější úkol se jeví odhalování organizovaných krádeží, protože se jedná o činnost zahrnující několik osob, kterou se snaží pachatelé této kriminální činnosti skrývat.

Tento druh projektu vyvolává i etické otázky, ve smyslu zneužití programu pro sledování lidí a podporu „Velkého bratra“ [22]. Tato práce zkoumá možnosti sledování lidí hlavně z teoretického hlediska, etickým otázkám se dále nevěnuje.

Metody pro analýzu a sledování objektů v obraze se používají také například pro rozpoznávání znakové řeči [31]. Další zajímavou oblastí může být automatický stříh videosekvencí na základě analýzy pohybu v části videa podobně jako v [4]. Tyto metody mohou řešit i otázky spojené s autonomním řízením robota na základě obrazu pořízeného ze senzorů a videokamer robota.

Se současným trendem masivního rozvoje internetového telefonování a videokonferencí

lze počítat i se zvyšováním počtu webkamer připojených k osobním počítačům. Tento jev je dalším impulzem k vývoji software pro interakci počítače s člověkem na základě obrazu pořízeného z webkamery, jako je například ovládání počítače pomocí gest rukou nebo výrazů obličeje. Dalším příkladem aplikace rozpoznávání objektů v obraze je virtuální a rozšířená realita, kde lze pro zlepšení kvality interakce člověka s počítačem využít haptických zařízení[46].

Problematiku virtuální reality lze chápat nejen z pohledu sledování a analýzy pohybu objektů ve scéně, ale i z pohledu rekonstrukce 3D scény na základě obrazu získaného z videokamery. Lze předpokládat, že problém sledování pohybu objektu ve scéně se z velké části překrývá s problémem vzájemného pohybu scény a videokamery.

Významný rozvoj je patrný i v oblasti vestavěných zařízení jako jsou mobilní telefony, digitální kamery nebo digitální fotoaparáty, ve kterých se používají metody zpracování obrazu pro zlepšení automatického zaostření nebo pro rozpoznávání piktogramů [34].

Ve 2. kapitole je popsáno získávání dat z obrazu jako celku. Jsou zde popsány základní přístupy pro získávání dat z videosekvencí a nastíněné souvislosti mezi jednotlivými metodami zpracování obrazu, strojového učení a statistického modelování. V následujících kapitolách jsou potom detailněji popsány jednotlivé metody.

Kapitola 3 obsahuje neúplný přehled základních metod předzpracování obrazu vhodných pro vyhledání objektu, zabývá se problematikou získávání digitálního obrazu a možnostmi reprezentace obrazu tak, aby to bylo vhodné pro získávání informací o pohybu objektů.

Kapitola 4 se zabývá možnostmi získávání informací o objektech v obraze na základě jednoho snímku videosekvence. Jsou zde zmíněny metody získávání informace z obrazu na základě dělení obrazu. Je zde popsán i bottom-up přístup, který se snaží o identifikaci jednoho objektu reprezentovaného několika samostatnými oblastmi obrazu. Nakonec je zde prezentován i přístup, který je založený na principu strojového učení a statistického modelování.

V kapitole 5 jsou popsány principy pro shrnutí informací z jednotlivých snímků na základě jejich souvislosti. Jsou zde prezentovány hlavní problémy při vyhledávání objektů ve videosekvenci a metody nejčastěji používané k těmto účelům. Jedná se o problémy vyhledání objektu ve dvou sousedních snímcích a problém překrytí objektu.

Následující kapitola se zabývá využitím zjištěných trajektorií. Jsou v ní popsány principy ověření správnosti dat získaných předcházejícími metodami, metoda pro analýzu trajektorií pomocí Skrytých Markovových Modelů a následně metoda pro vyhodnocení komplexních událostí.

Kapitola 7 se zabývá způsobem implementace a zabudováním algoritmů pro sledování objektů do programových systémů. Jsou zde diskutovány možné způsoby předávání dat a rozhraní pro programování aplikací. Je zde diskutován také konkrétní způsob implementace a výsledky dosažené při jejím testování.

Kapitola 2

Získávání znalostí z videosekvencí

Znalost lze definovat jako atomickou hodnotu, která přímo popisuje nějaký jev v obraze. Získávání znalostí je pojem, který označuje proces zahrnující přípravu dat, integraci, dolování z dat a zobrazení znalostí. Dolování dat je nejdůležitější část tohoto procesu. Je to pojem, který je definován jako netriviální proces vyhledání správných, neobvyklých, potenciálně užitečných a jednoznačných vzorů v datech [12]. Tato definice se pro účely zpracování video sekvencí často podle [12] a [40] rozšiřuje na dolování dat z videosekvencí pomocí hledání podobností nebo pomocí dalších automatizovaných technik pro detekci událostí, rozpoznávání optického charakteru obrazu nebo analýzy řeči.

Jako vstupní data tedy chápeme videosekvenci a definici požadovaných znalostí, jako výstupní data chápeme požadovanou znalost o videosekvenci. V závislosti na zadaném úkolu se liší míra abstrakce znalosti. V případě videozáznamů z hromadné dopravy lze požadovanou výstupní znalost definovat například jako výskyt jevu přeskakování přes turnikety ve scéně. Výstup lze potom definovat jako dobu, kdy došlo ke sledovanému jevu.

V této kapitole jsou popsány základní přístupy používané pro získávání dat z videosekvencí, které jsou relevantní pro tuto diplomovou práci. Jsou zde popsány obecné principy, které lze aplikovat na většinu níže popsaných metod pro zpracování obrazu, strojového učení a statistického modelování.

2.1 Přímé získávání znalostí

Přímá metoda získávání dat z videosekvencí spočívá v aplikaci jednoho algoritmu na videosekvenci jehož výsledkem jsou přímo požadované údaje. Tento algoritmus může zahrnovat několik metod pro zpracování obrazu, které jsou na sobě závislé tak, že není možné je použít samostatně.

Tento přístup poskytuje obvykle nízkou míru abstrakce. Zdrojový kód jednotlivých metod je vzájemně velmi úzce provázaný. Naproti tomu tento přístup umožňuje velkou optimalizaci na úrovni komunikace a předávání dat mezi jednotlivými kroky metody. Nevýhodou je menší znovupoužitelnost jednotlivých částí programu.

Jednotlivé metody pro zpracování obrazu a dolování dat z obrazu jsou popsány v následujících kapitolách.

2.2 Blokový přístup

Blokový přístup na rozdíl od přímého přístupu kombinuje několik metod zpracování obrazu a strojového učení pro získání požadované informace. Cílem je vytvoření orientovaného grafu, kde každý uzel reprezentuje jeden blok. V každém bloku se provede jedna operace. Obvyklý postup je ten, že se v jednotlivých blocích snižuje množství informace, kterou je obraz reprezentovaný a následně pomocí metod strojového učení získávají informace o zajímavých jevech v obraze. Mezi jednotlivé funkční bloky se obvykle vkládají bloky pro filtraci šumu fungující jako postprocessing předcházející metody a předzpracování pro následující metodu.

Jeden funkční blok může realizovat několik operací a chápeme ho jako jednu aplikaci „Přímého přístupu“ popsaného v předcházející podkapitole. Jednotlivé bloky musí mít dobře definované rozhraní pro vzájemné předávání dat. Tento přístup umožňuje kombinování jednotlivých bloků pro dosažení optimálního výsledku pro dané vstupní data a dekompozici problému na několik dílčích podproblémů.

Použití přímého přístupu je ekvivalentní zpracování pomocí blokového přístupu s jedním funkčním blokem. Zpracování obrazu z videosekvence v jednom funkčním bloku může být dobře optimalizované pomocí paralelního zpracování na úrovni, která je blízko ke struktuře a způsobu uložení dat. Akcelerace zpracování možná i na úrovni paralelního zpracování některého z funkčních bloků.

Pomocí těchto funkčních bloků lze znázornit i kombinaci několika metod. V případě, že je jedna metoda pomalejší, ale získává kompletní informace o obraze, a druhá metoda je rychlejší, ale potřebuje inicializaci, tak lze toto zpracování znázornit jako paralelní větev.

2.3 Hierarchický přístup

Hierarchický přístup umožňuje ještě větší míru abstrakce než jednoduchý blokový přístup. Jednotlivé funkční bloky je možné do sebe zanořovat a vytvářet tak jejich hierarchii.

Ukázka hierarchického přístupu je znázorněna na obrázku 2.1. Na nejvyšší úrovni vidíme, že v obraze získaného z průmyslové kamery probíhá vyhledání potenciálně zajímavých objektů v jednotlivých snímcích, ve druhém kroku probíhá sledování těchto objektů v čase. Následně se informace o pohybu objektů analyzuje a v posledním kroku je znalost získaná z videosekvence prezentována operátorovi.

V případě pohledu na nižší úroveň abstrakce a do detailu jednotlivých funkčních bloků lze zjistit jakým způsobem jsou realizované jednotlivé operace. Pro vyhledání objektu pohybujícího se ve scéně snímané staticky umístěnou kamerou lze využít metody pro odečítání pozadí, které vytvářejí model pozadí z většího úseku obrazu, a metody pro hledání pohybu ve videosekvenci, která je založena na vytváření krátkodobého modelu pozadí. Vhodnou kombinací těchto metod se získá informace o místech s nejpravděpodobnějším výskytem jednotlivých objektů pro každý snímek.

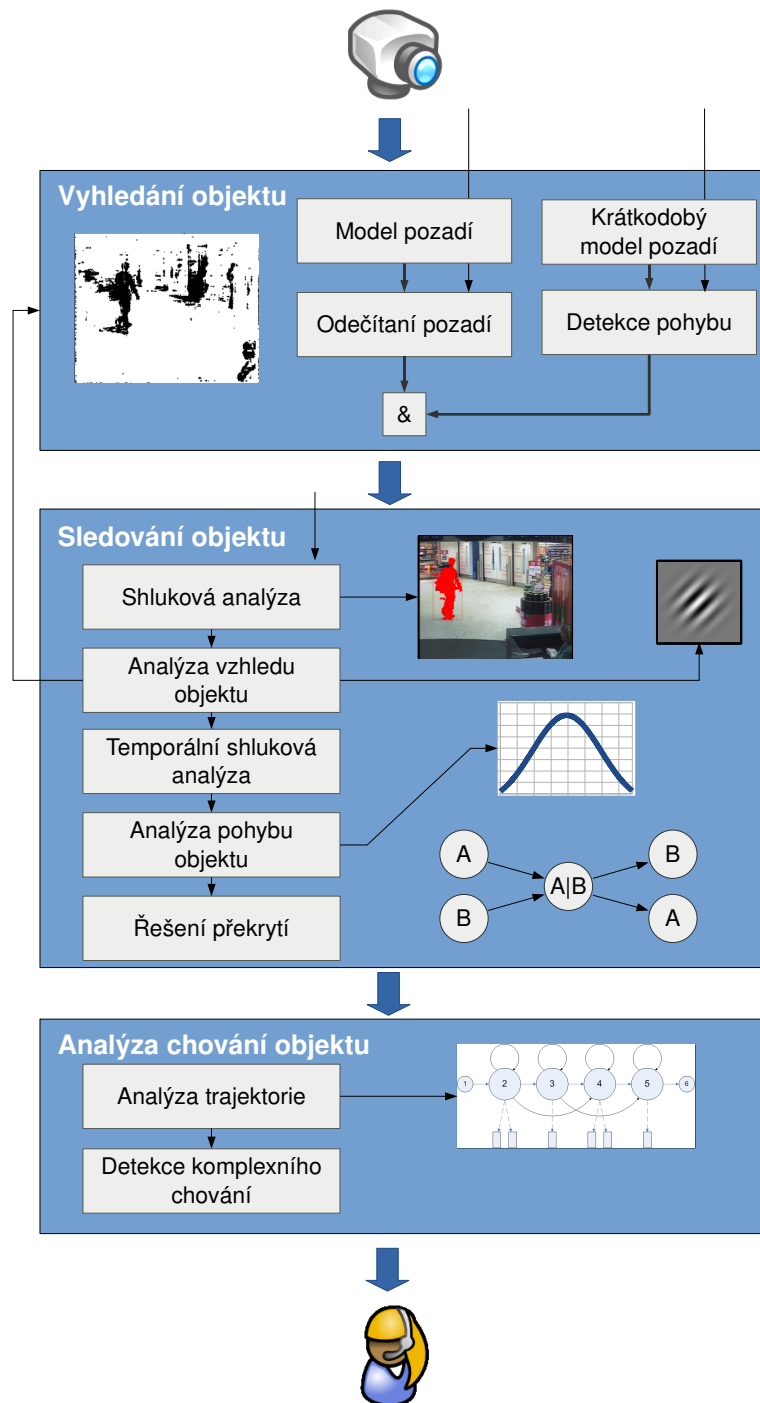
Ve druhém kroku se v obraze popisujícím oblasti s vysokou pravděpodobností výskytu objektů pohybujících v popředí scény naleznou souvislé oblasti omezené určitými kritérii, přičemž se předpokládá, že každá z těchto oblastí popisuje jeden nezávislý objekt. Tyto oblasti se analyzují a následně popíší pomocí nějakého modelu vzhledu. Model vzhledu se může skládat z několika metod. Objekt může být popsán například pomocí texturních příznaků nebo podle tvaru. Blok naznačený jako temporální shluková analýza se spolu s analýzou pohybu objektu snaží nalézt souvislost mezi polohou jednotlivých objektů v sousedních snímcích, kde první z těchto bloků se zabývá pouze sousedními snímky a druhý z těchto

bloků se zabývá analýzou pohybu z dlouhodobějšího hlediska. Poslední část bloku pro sledování objektu je blok zabývající se překrytím objektu. Tato část řeší problém daný 2D reprezentací třírozměrného prostoru. Problém lze popsat tak, že v jedné souvislé oblasti pravděpodobného výskytu objektu se nachází více než jeden objekt. Tyto objekty je nutné oddělit na základě modelu vzhledu objektu a dlouhodobého pohybu objektu. V případě, že není možné objekty rozlišit na základě žádného z těchto modelů je nutné předpokládat, že došlo úplnému překrytí objektů. Tento stav je na obrázku ilustrován orientovaným grafem. K situacím, ve kterých lze pozorovat úplné překrytí objektů, dochází v hromadné dopravě poměrně často.

Ve třetím kroku se zpracovává informace o pohybu objektu. Pohyb objektu může být vyjádřen například počáteční a koncovou polohou objektu, rychlostí objektu nebo změnou velikosti objektu. Pohyb objektu může být popsán pomocí trajektorie. Obtížným problémem analýzy pohybu je způsob převodu trajektorie na znalost. V posledním kroku analýzy chování objektu se informace získané různými přístupy zpracovávají a kombinací několika jednodušších znalostí o chování objektu vytvářejí komplexní znalosti o chování objektu.

V posledním kroku dochází k ukládání znalostí získaných zpracováním videa a jejich prezentování uživateli systému.

V rámci zachování relativní jednoduchosti znázornění přístupu k problému jednotlivé dílčí bloky popisují opět hierarchii metod, které se používají pro zpracování obrazu, statistické modelování a strojové učení. Každý z dílčích bloků se skládá z několika metod. Nejčastější operace probíhající v těchto blocích jsou filtrace obrazu jako preprocessing a postprocessing jednotlivých algoritmů.



Obrázek 2.1: Schéma zpracování obrazu

Kapitola 3

Zpracování obrazu

Zpracování obrazu je velmi rozsáhlá oblast patřící k vědnímu oboru zpracování signálů. Většina metod pro zpracování obrazu přistupuje k obrazu jako ke dvourozměrnému signálu, na který aplikuje metody zpracování signálu. Zpracování obrazu lze rozdělit na digitální a analogové zpracování obrazu. Tato práce se zabývá pouze digitálním zpracováním obrazu. Obrazy z analogových zařízení lze převést pomocí AD převodníků do digitální podoby a následně provést digitální zpracování.

Dvou rozměrný digitální signál je diskrétní reprezentací scény snímané kamerou. Tento signál je definovaný vzorkovací frekvencí, jednotlivými vzorky a jejich amplitudou. V případě digitálního obrazu každý vzorek reprezentuje jeden pixel obrazu a amplituda reprezentuje jeho odstín a barvu. Barvu pixelu v obrazu lze reprezentovat různými způsoby, možné reprezentace jsou popsány v podkapitole 3.1. Další informace vztahující se k digitálnímu zpracování signálů popsány v této kapitole byly čerpány z [43] a [20].

Vstupními daty pro zpracování obrazu jsou obvykle fotografie nebo jednotlivé snímky videosekvence. Výstupem zpracování obrazu je buď upravený obraz nebo data reprezentující nějakou vlastnost obrazu.

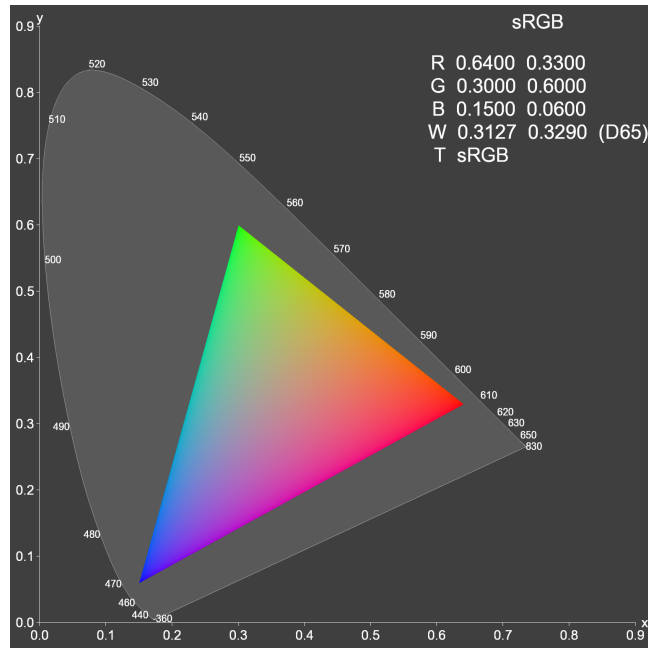
V této kapitole jsou uvedeny některé metody pro zpracování obrazu, které jsou relevantní k řešenému problému. Bližší informace k metodám popsáným v této kapitole a další metody zpracování obrazu lze nalézt v [48] a [16]. Analýzou dat, vyhledáním a identifikací objektů v obraze se věnuje následující kapitola.

3.1 Barevné prostory

Barva je definovaná jako vjem, který vytváří světlo dopadající na sítnici lidského oka. Viditelné světlo je definované jako světlo o vlnové délce přibližně mezi 400 a 800nm. Jednotlivé snímací zařízení (lidské oko nebo kamera) jsou schopné snímat jenom světlo o určitých vlnových délkách. Dosažitelnou oblast barev je definovaná jako gamut. Ukázka barev dosažitelných běžným CRT monitorem je na obrázku 3.1. Jednotlivé obrázky popisující gamut a ukázky barevných prostorů byly převzaty z [39].

Pro diskrétní reprezentaci barvy se používají různé barevné prostory v závislosti na typu zařízení, které s barvou pracuje, a na způsobu použití. Jednotlivé barevné prostory jsou pro ilustraci zobrazeny na obrázku 3.2. Zařízení jako monitor a videokamera obvykle pracují s aditivní reprezentací barvy, protože je to přirozené z hlediska použité technologie. Aditivní reprezentace barvy se popisuje barevným prostorem RGB¹. Naproti tomu zařízení jako

¹Red Green Blue



Obrázek 3.1: Typický CRT gamut RGB

tiskárny pracují obvykle se subtraktivní reprezentací barvy, pro kterou se používá barevný prostor CMYK².

Podle Alvy Ray Smithe [39] jsou pro lidské vnímání přirozenější barevné modely pracující s jasovou složkou obrazu. Příkladem těchto barevných prostorů je HSV³ (a k němu komplementární HSL⁴) nebo YUV⁵ a YCbCr⁶ než model RGB. Předpokládá se, že některé metody zpracování obrazu a počítačového vidění kopírující způsob lidského vnímání okolního světa mohou mít lepší výsledky, pokud se provedou nad barevnými modely popisující lépe jasovou složku obrazu.



Obrázek 3.2: Barevné modely - zleva RGB, CMYK, HSV/HSL

Jednotlivé reprezentace barvy je možné vzájemně převádět, například pro převod z barevného prostoru RGB do barevného prostoru YUV platí vztah 3.1.

²Cyan Magenta Yellow Key (key je černá barva)

³Hue Saturation Value

⁴Hue Saturation Lightness

⁵Y = luma, UV je informace o barvě

⁶Y = luma, Cb = Chroma Blue, Cr = Chroma Red

$$\begin{bmatrix} Y' \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.1)$$

HMMD

Méně známým barevným prostorem je HMMD⁷. Podle [19] je tento barevný prostor nejvhodnější pro dolování dat z videa, protože se jeho reprezentace blíží lidskému vnímání více, než je tomu například u barevného modelu HSV nebo YUV.

Barevná složka *Hue* reprezentuje zabarvení a je možné ji popsat úhlem od 0 do 360 stupňů, podobně jako v modelu HSV (viz. obrázek 3.2 vpravo). Složka *Max* popisuje množství černé barvy v obraze a charakterizuje clonu. Složka *Min* popisuje míru bílé barvy v obraze a lze jí popsat například jako zabarvení. Barevná složka *Diff* popisuje míru šedé barvy v obraze a její vzdálenost od „čisté“ barvy. Tuto složku lze charakterizovat jako odstín barvy. Barevný model HMMD pracuje také se složkou *Sum*, která popisuje jasovou složku obrazu. Nevýhodou tohoto barevného modelu je určitá redundance (pro popis barvy postačuje trojice parametrů *Hue*, *Max* a *Min* nebo *Hue*, *Diff* a *Sum*).

Převod z barevného prostoru RGB do HMMD lze popsat rovnicí 3.2.

$$\begin{aligned} max &= Max(r, g, b) \\ min &= Min(r, g, b) \\ diff &= max - min \\ sum &= (max + min)/2 \end{aligned}$$

$$h = \begin{cases} 0 & \text{if } max = min \\ 60^\circ \times \frac{g-b}{max-min} + 0^\circ, & \text{if } max = r \text{ and } g \geq b \\ 60^\circ \times \frac{g-b}{max-min} + 360^\circ, & \text{if } max = r \text{ and } g < b \\ 60^\circ \times \frac{b-r}{max-min} + 120^\circ, & \text{if } max = g \\ 60^\circ \times \frac{r-g}{max-min} + 240^\circ, & \text{if } max = b \end{cases} \quad (3.2)$$

Barevný model HMMD je součástí MPEG-7 jako součást deskriptoru objektu, přičemž modeluje rozložení lokálně pro určitou část obrazu [21].

3.2 Fitrace

Metody fitrace obrazu jsou založeny principech zpracování signálu. Základní rozdělení digitálních filtrů je na FIR⁸, tedy filtry s konečnou impulsní odezvou a IIR⁹ s nekonečnou impulsní odezvou.

Cílem fitrace obrazu je nejčastěji odstranění nebo zvýraznění části obrazu. Typickým příkladem části obrazu, která je nežádoucí, a proto je snaha zbavit se jí, je šum. Fitrace šumu je operace, kterou lze provést téměř v libovolném kroku zpracování obrazu. Jejím cílem je eliminování systematických chyb měření a výpočtu. V ideálním případě obraz neobsahuje žádný šum a není potřeba provádět filtraci obrazu. V praxi ale dochází k mnoha jevům, které výrazně ovlivňují účinnost metod pro strojové zpracování obrazu a počítačové vidění. Kvalita obrazu z kamery je ovlivněná například kvalitou optické soustavy kamery, CCD

⁷Hue Max Min Diff

⁸Finite Impulse Response

⁹Infinite Impulse Response

čipu, světelnými podmínkami ve scéně, nastavením clony, doby expozice, typem a stupněm komprese obrazu nebo případně elektrostatickými jevy při přenosu analogového obrazu.

Konvoluce

Konvoluce je metoda lineární filtrace obrazu. Je definovaná jako filtrace obrazu realizující lineární funkci. Lineární funkce je definována jako funkce, u které platí pro všechny hodnoty jednotlivých pixelů v obraze vztah 3.3.

$$\forall x, y : f(o_1, x, y) + f(o_2, x, y) = f(o_1 + o_2, x, y) \quad (3.3)$$

Lineární konvoluční filtr je filtr popsaný konvolučním vzorem. Výsledek filtrace se počítá pixel po pixelu s tím, že každý pixel se vypočítá jako konvoluce okolí daného pixelu. Protože se jedná o diskrétní obrazy, lze konvoluci nahradit sumou. Obvyklá velikost konvolučního vzoru bývá 3x3, 5x5 nebo 7x7 pixelů. Výpočet konvoluce pro konvoluční jádro k o velikosti 3x3 pixelů, vstupní obraz x a výstupní obraz y je popsán rovnicí 3.4.

$$y_{(x,y)} = \sum_{ii \in I} \sum_{jj \in J} x_{(i+ii, j+jj)} k_{ii, jj}, I = \{-1, 0, 1\}, J = \{-1, 0, 1\} \quad (3.4)$$

Pomocí konvoluce lze snadno realizovat například hranový detektor, filtry pro zaostření nebo rozmazání obrazu.

Mediánový filtr

Mediánový filtr je typický příklad nelineárního filtru, který se používá například k filtraci výstřelového šumu nebo šumu typu pepř a sůl. Pro nelineární filtr obecně platí, že nespĺňuje podmínku linearity, definovanou výrazem 3.3. Tento typ filtru je postavený na principu výběru vhodné hodnoty z okolí pro daný pixel.

Morfologické operace

Matematická morfologie je definovaná jako teorie pro analýzu prostorových struktur. Jejím cílem je analýza tvaru a vzhledu prostorových struktur[16]. Morfologické operace lze aplikovat jako nelineární filtr na binární obraz.

Binární morfologie je operace, kterou lze definovat nad bodovými množinami. Obraz lze modelovat pomocí bodových množin libovolné dimenze. Morfologická transformace je dána relací mezi bodovou množinou definovanou obrazem O a menší bodovou množinou definovanou strukturálním elementem E . Aplikace morfologické operace na obraz O odpovídá systematickému posunu strukturálního elementu E po obraze O .

Jednou z morfologických operací je binární dilatace. Tato operace je definována jako sčítání dvou bodových množin 3.5 a následné sjednocení jejich posunutých obrazů 3.6.

$$O \oplus E = \{p \in \mathbb{Z}^2 : p = x + b, x \in O \wedge b \in E\} \quad (3.5)$$

$$O \oplus E = \bigcup_{b \in E} O_b \quad (3.6)$$

Komplementární operace k binární dilataci je binární eroze. Tato operace je definovaná vztahem 3.7.

$$O \ominus E = \{p \in \mathbb{Z}^2 : p = x + b \in O \wedge \forall b \in E\} \quad (3.7)$$

Kombinací morfologických operací binární dilatace a eroze lze vytvořit operace binárního otevření a binárního uzavření. Tyto metody lze aplikovat pro filtrování šumu v binárním obraze [10]. Další podrobnosti o morfologických operacích lze nalézt v [2] a [16].

Prahování

Prahování je metoda filtrace obrazu, která nastaví novou hodnotu obrazu na základě amplitudy původního signálu pro $\forall p \in O(x, y)$, kde O původní obraz, platí 3.8. Přičemž výsledkem této operace je většinou binární obraz, kde $A = 0$ a $B = 1$. Parametr T určuje nastavení prahu filtrace.

$$f(p) = \begin{cases} A & \text{pokud } p < T \\ B & \text{pokud } p \geq T \end{cases} \quad (3.8)$$

Základním předpokladem této metody je, že hledané oblasti mají vyšší hodnotu jasu než je práh T . Tento práh lze definovat staticky pro celý obraz jako je to v 3.8. Tato metoda obvykle doplňuje některou jinou metodu pro zpracování obrazu, například metodu odečítání pozadí.

Prahování se nejčastěji provádí na barevném nebo šedotónovém obraze, ale lze jej provést například i na abstraktním obraze definujícím pravděpodobnost výskytu jevu v daném pixelu obrazu. V případě barevného obrazu se prahování provádí obvykle pro každou barevnou složku zvlášť. Nejvhodnější úroveň prahu lze určit buď ručně (metodou pokus-omyl) nebo automaticky. Ruční nastavení hodnoty prahu v dynamicky měnícím se obraze může být problém. Pro automatické nebo poloautomatické zvolení nejvhodnějšího prahu se používá barevný nebo šedotónový histogram obrazu. Nejjednodušší metoda volby prahu je pomocí střední hodnoty nebo mediánu odstínu všech pixelů v obraze.

Modifikace prahování

Modifikace prahování vyžadují většinou automatický nebo poloautomatický přístup pro nastavení prahu.

V případě, že se požaduje nalezení lokálních extrémů v obraze, tak lze použít *lokální adaptivní prahování*, přičemž obraz se rozdělí po blocích o zvolené velikosti a práh se nastavuje pro každý blok. Pokud má metoda lokálního adaptivního prahování nastavenou velikost bloku na jeden pixel, tak se práh nastavuje pro všechny pixely obrazu zvlášť. Toto prahování nazývá *adaptivní dynamické prahování*.

Další variantou je *prahování s více prahy*, kdy se staticky zvolí n prahů, vůči kterým se porovnává každý pixel obrazu a výsledkem je n -ární obraz. Tento přístup je možné kombinovat s adaptivním lokálním i adaptivním dynamickým prahováním.

3.3 Geometrické transformace

Protože kamera snímá obraz perspektivně a na čočce dochází k deformaci obrazu, tak je nutné provádět korekce obrazu. Při sledování objektů lze pozorovat změny tvaru objektu. Pro popis těchto změn tvaru objektu se snažíme najít zobrazení, které by odpovídalo těmto změnám. Toto zobrazení lze popsat pomocí rovnice 3.9

$$T(u, v) = [x(u, v), y(u, v)] \quad (3.9)$$

Geometrické transformace lze rozdělit na lineární geometrické transformace, mezi které patří posunutí, rotace, změna měřítka a zešikmení. Dále na nelineární geometrické transformace jako jsou například wrapping a morphing. Mezi nelineární geometrické transformace patří také transformace související s projekcí a změnou perspektivy.

Separabilní transformace

Dělitelnost transformace je vlastnost, kterou lze definovat jako schopnost vytvořit transformaci skládáním jednodušších transformací, tak jak je to popsáno v rovnici 3.10. Skládání jednotlivých transformací umožňuje vytvoření jedné transformace postihující všechny operace, které je potřeba nad obrazem provést.

$$T(u, v) = F(u, G(v)) \quad (3.10)$$

Afinní transformace

Afinní zobrazení (transformace) je speciální případ geometrického zobrazení, při kterém zůstává zachována rovnoběžnost přímek. Afinním zobrazením lze popsat posunutí, otočení, změnu měřítka, zkosení nebo operaci vzniklou jejich skládáním. Afinní transformace lze popsat vztahem 3.11, kde transformace je dána maticí A a vektorem b .

$$x \mapsto Ax + b \quad (3.11)$$

V případě použití homogenních souřadnic lze například rotaci podle osy Z zapsat vztahem 3.12, kde γ je úhel natočení obrazu. Uvedenou matici transformace lze použít pro rotaci bodu ve dvou rozměrném i tří rozměrném prostoru. Díky uniformnímu tvaru a vlastnosti separability lze jednotlivé operace snadno skládat násobením matic.

$$X' = X \begin{vmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (3.12)$$

Vstupními daty pro geometrickou transformaci je obraz (pole bodů) a výstupem je obraz (pole bodů). Výstupní obraz má obvykle jinou velikost než vstupní obraz. Detailnější popis Afinních transformací je možné nalézt v [54]. Problematiku počítání s perspektivou a deformací čočky lze nalézt v [6].

Další možné využití geometrických transformací je například při stabilizaci obrazu kamery [25].

3.4 Extrakce příznaků

Příznaky lze definovat jako parametry popisující nějaký aspekt obrazu a lze je popsat jako průmět obrazu do prostoru příznaků. Metody zpracování obrazu popsané v předcházejících podkapitolách lze chápat také jako extrakci příznaků. Příznaky se používají v dalších metodách počítačového vidění, například pro rozpoznání objektu. Důvodem používání příznaků je jejich schopnost lépe popsat hledaný jev v obraze, čímž umožňují lepší flexibilnější přístup k problému.

Příznaky popisující tvar

Jedním z nejdůležitějších parametrů objektů v obraze je tvar těchto objektů. Tento tvar je možné reprezentovat pomocí šablony, například maskou popisující siluetu objektu. Opačný přístup popsat objekt pomocí rovnice, například pomocí B-Spline, tak jak je to popsáno v [18]. Pro nalezení příznaků popisujících tvar objektu je možné použít například následující metody:

- Houghova transformace
- Local rank differences
- metody pro detekci hran (Cannyho, Sobelův, Harrisův detektor)
- metody pro detekci rohů (Laplacian Gaussianu, Harissův detektor)
- metody pro detekci blobů (MSER, Grey-level blobs)

Například metodu detekce hran pomocí Sobelova detektoru [39] lze popsat jako nelineární filtraci vytvořenou kombinací dvou konvolucí, tak jak je to popsáno v rovnici 3.13

$$O_{edges} = \sqrt{\left(\left(\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * O_{in}\right)^2 + \left(\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * O_{in}\right)^2}\quad (3.13)$$

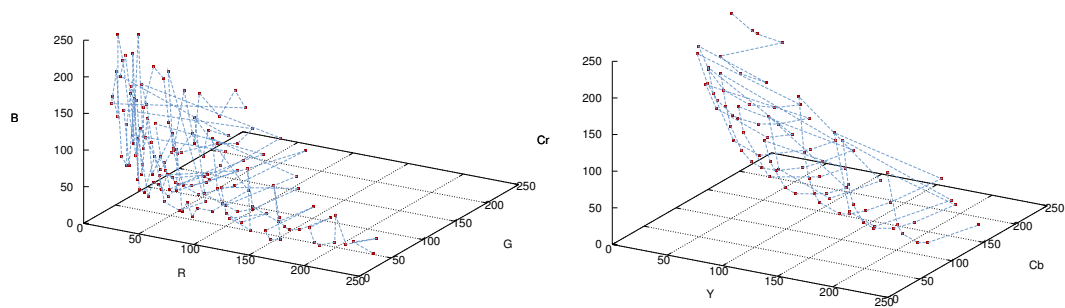
Texturní příznaky

Textura je nástroj pro popis povrchu objektu. Je definovaná jako opakující se struktura primitiv tzv. texelů. Texel popisuje charakteristiky textury jako je podobnost tvarů v textuře, rozložení intenzit, orientaci a velikost, přičemž platí, že počet těchto primitiv je mnohem větší než jejich variabilita. Pro detekci textury je potřeba definovat její charakteristické měřítko. Podle dekomponovatelnosti a výraznosti textury se rozlišují silné a slabé textury. Detailnější informace o zde popisovaných texturních příznacích je možné nalézt v [52].

Texturu objektu lze popsat například pomocí lokálních barevných histogramů. Histogram je definovaný jako četnost výskytu určité hodnoty. V případě barevného histogramu se jedná o četnost výskytu určité barvy v obraze. Příklad barevného histogramu je znázorněn na obrázku 3.3. Hodnoty jednotlivých binů¹⁰ lze znázornit například jasem nebo velikostí bodu v prostoru. V tomto případě jsou v obrázku znázorněny pouze nejdůležitější biny histogramu (všechny stejnou barvou i velikostí). Pro lepší orientaci v prostoru jsou nejbližší body spojené čarami, přičemž tato vzdálenost je definována po vrstvách. Body jsou tedy spojeny tak, jak byly postupně vykresleny. Na obrázku jsou ukázány barevné histogramy ve 3D pro stejný objekt v různých barevných prostorech. Vlevo je histogram pro barevný prostor RGB, vpravo je histogram pro barevný prostor YCbCr.

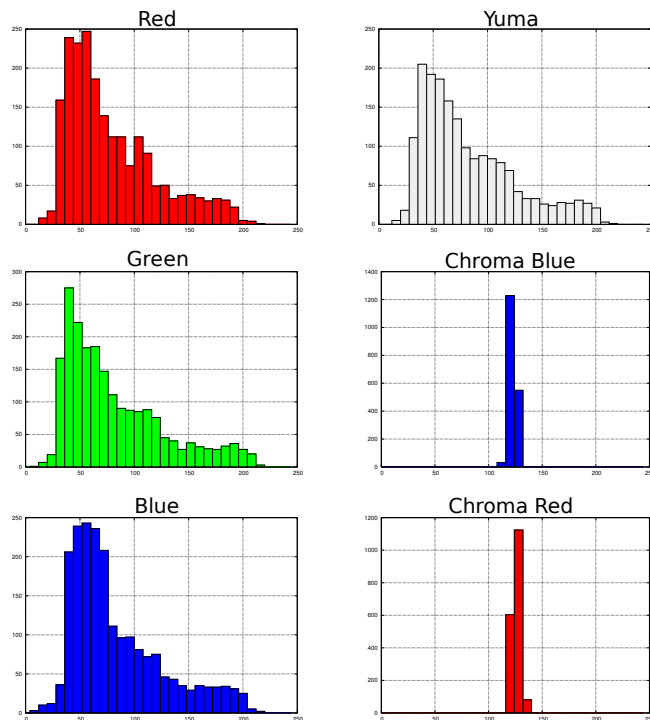
Třírozměrná reprezentace histogramu může být poměrně hodně paměťově náročná a zbytečně komplikovaná pro další zpracování obrazu a interpretaci obrazových dat, proto se pro zobrazení často používají průměty do dvourozměrného prostoru. Zobrazení dvourozměrného průmětu výše uvedeného barevného histogramu je ukázáno na obrázku 3.4. Vlevo jsou zobrazeny průměty třírozměrného histogramu v barevném prostoru RGB, vpravo v barevném prostoru YCbCr. Z dvourozměrných průmětů histogramů se na první pohled zdá,

¹⁰jednotlé sloupce histogramu



Obrázek 3.3: 3D Barevný histogram

že takovýto průběh dvourozměrného histogramu by mohl být více diskriminativní a mohl by pomoci lépe odlišit jednotlivé objekty.



Obrázek 3.4: Průmět barevného histogramu do 2D prostoru

Barevné histogramy byly vytvořeny z jedné oblasti ve videosekvenci ukázané na obrázku 3.5. Tento obrázek popisuje část hlavy člověka.

Další metody vhodné pro popis textury, které jsou za poměrně robustní, jsou metody založené na CAR¹¹ příznacích, na Local Binary Patterns [17] a modelování textury pomocí Gaborových vlněk [51]. Metoda modelování textury pomocí CAR příznaků je založena na Gaussovských pyramidách a Markovových náhodných polích. Její přesnější popis a po-

¹¹Causal Autoregressive Random texture model



Obrázek 3.5: Část hlavy člověka – obrázek použitý pro demonstraci tvorby histogramů

rovnání s dalšími metodami pro popis textur lze nalézt v [32]. Dalšími metodami pro popis textur relevantními pro problém sledování objektů v obraze mohou být například:

- Histogramy korekurence
- Radiální příznaky (na základě Výkonového spektra)
- Autokorelační funkce
- Local Binary Patterns

Pro robustnější popis určitých typů objektů měnících se v čase lze rozšířit texturu o temporální rozměr „Motion Texture“ [23]. Typickým příkladem tohoto typu objektu jsou listy stromu pohybující se ve větru.

Kapitola 4

Počítačové vidění po snímcích

Počítačové vidění je vědní obor zabývající viděním počítačů. Pojem vidění se obvykle považuje za interpretaci obrazu a následné vytvoření modelu scény. V širším slova smyslu nemusí jít o zpracování obrazu, ale o vytvoření modelu okolí na základě libovolného vnímání okolního světa, například pomocí sonaru. Tato práce se zabývá vytvořením modelu scény založeného na interpretaci obrazových dat získaných z jednotlivých snímků videosekvence.

Automatické vytvoření globálního popisu scény může být poměrně náročný úkol, protože scény ve skutečném světě se často mění v čase. Pro další automatické zpracování se obvykle definuje podmnožina scény, která obsahuje potencionálně zajímavé informace z pohledu dalšího zpracování. Takovou podmnožinou scény mohou být libovolné objekty. K nejčastěji vyhledávaným objektům ve scéně patří člověk. Jeho poloha a póza může být popsána například pomocí polohy rukou a obličeje [51].

Pro vyhledání objektů v obraze je nutné nejdříve definovat jakým způsobem se budou popisovat nalezené objekty. Objekty lze popsat například jako binární obraz, který přesně vystihuje tvar nalezeného objektu pro daný snímek. Nevýhodou tohoto přístupu je množství dat, které popisují objekt. V případě, že je možné popsat objekt trojrozměrně, tak lze použít objemovou reprezentaci, která se často používá například v medicíně pro uložení dat z počítačového tomografu nebo magnetické rezonance. Nejjednodušší možnou reprezentací objektu v prostoru je jeden bod. Při popisu tímto způsobem dochází, ke ztrátě poměrně velkého množství informací. Nejvhodnější bod pro reprezentaci tělesa je pravděpodobně jeho těžiště.

Další možnou reprezentací objektu je jeho popis pomocí obalového tělesa. Z důvodu jednoduchosti popisu se ve dvourozměrném prostoru jako obalové těleso nejčastěji používá obdélník. Ve třírozměrném popisu lze jako obalové těleso použít povrchovou reprezentaci objektu popsanou jeho polygony. Jednoduššími strukturami používanými v reálných aplikacích pro třírozměrný prostor jsou kvádr a koule.

V případě, že polohu objektu lze stanovit jenom jako pravděpodobnost jeho výskytu s normálním rozložením, lze polohu popsat Gaussovou křivkou. Použitý popis objektu závisí hlavně na tom, jakým způsobem se bude tento popis dále využívat. Je nutné zvolit kompromis mezi množstvím uložené informace, jednoduchostí a přesností reprezentace objektu.

Tato kapitola se zabývá počítačovým viděním na úrovni jednoho snímku ve videosekvenci. Jsou zde rozebrány některé metody relevantní pro tuto diplomovou práci. Informace uvedené k jednotlivým metodám byly čerpány hlavně z [26], [30], [5], [11] a [16].

4.1 Segmentace obrazu

Segmentace je definována jako proces dělení obrazu na části, přičemž jednotlivé části odpovídají různým částem scény. Cílem segmentace je rozdělit obraz tak, aby bylo možné odlišit jednotlivé objekty ve scéně. Tento cíl je však často nedosažitelný, takže se obraz dělí na významné části, které jsou později analyzovány a interpretovány.

Oblasti zájmu

Jedna z možných metod filtrování obrazu je pomocí oblastí zájmu. Pokud se u některé ze zvolených metod zjistí, že má správnou odezvu jenom pro určitou část obrazu a nesprávnou odezvu pro zbývající část obrazu, popřípadě vyžaduje odlišné nastavení parametrů pro různé části obrazu, tak je pro metody pracující s obrazem lokálně možné obraz rozdělit na několik částí, které se zpracují odděleně. Některé části je možné odstranit z dalšího zpracování úplně.

Odečítání pozadí

Odečítání pozadí je jedna z nejjednodušších metod pro segmentaci obrazu. Tato metoda je založena na předpokladu, že je možné popsat pozadí scény a následně jej odstranit z obrazu.

Z hlediska interpretace světa, případně jeho třírozměrné reprezentace, lze pozadí chápat jako scénu, ve které se nevyskytují objekty zájmu (tj. objekty, které se mají vyhledat). V reálných situacích často není třírozměrná reprezentace scény dostupná a objekty popředí nejsou vždy úplně jednoznačně definovány. Pozadí scény je možné definovat i dvourozměrně jako průmět scény do roviny kamery a vytvářet model pozadí pouze pro tento průmět.

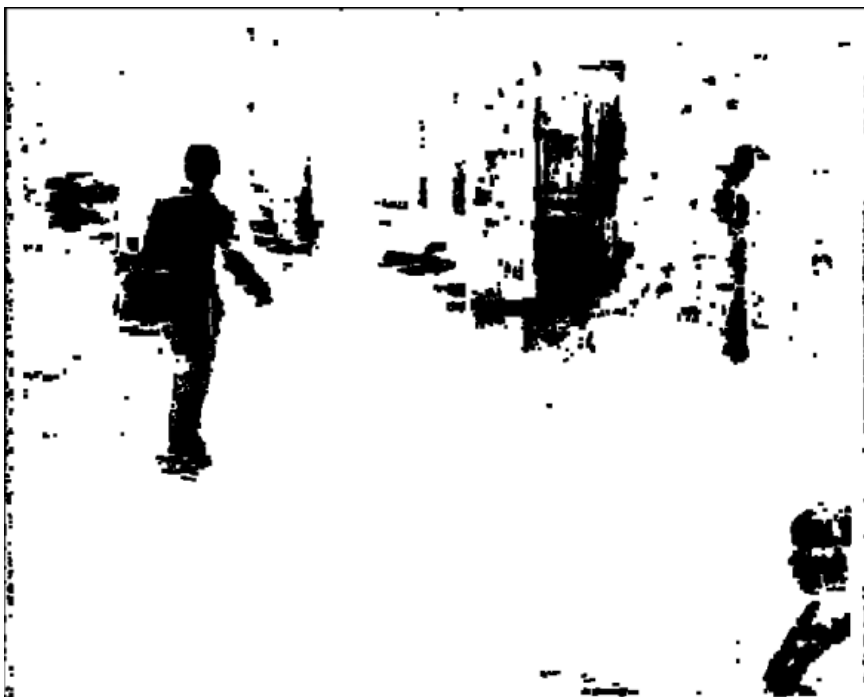
Jedním z problémů většiny reálných scén je neexistence konstantních podmínek. Nejvýrazněji se obvykle mění světelné podmínky scény. Tyto světelné podmínky je nutné modelovat jako součást pozadí scény a kompenzovat jejich vliv při detekci objektů popředí. Třírozměrná reprezentace pozadí scény předpokládá, že je možné oddělit tvar a texturu objektu, a po odečtení objektů pozadí získat korektní popis objektů popředí. V případě dvourozměrné reprezentace scény a jejího pozadí je kompenzování vržených stínů a světelných odlesků ještě obtížnějším problémem. Řešením tohoto problému je modelování pozadí včetně možných změn světelných podmínek. Protože barva textury objektu se vlivem vržených stínů liší hlavně v jasové složce, ale jenom minimálně v tónu barvy, tak lze s výhodou použít vhodný barevný model, tak jak to bylo popsáno v kapitole 3.1.

Dvourozměrný model pozadí se obvykle vytváří tak, že se postupně zpracuje několik anotovaných snímků a v případě, že daný pixel nepopisuje objekt popředí, tak se uloží hodnota barvy daného pixelu.

Dlouhodobý model pozadí

Jedním z hlavních problémů při vytváření modelu pozadí je odlišení objektů popředí od pozadí scény. Pro definování pozadí videosekvence, lze za určitých podmínek předpokládat, že v dlouhé videosekvenci se pro každý pixel vyskytuje barva pozadí s normálním rozložením hustoty pravděpodobnosti a jiným normálním rozložením barva objektů popředí. Pro správnou funkčnost této metody musí být střední hodnota těchto jevů rozlišitelná.

Na základě tohoto předpokladu lze pro každý pixel obrazu modelovat pravděpodobnost, že barva pro daný pixel patří nebo nepatří k pozadí scény. Příklad modelu popředí získaného na základě dlouhodobého modelu pozadí je znázorněn na obrázku 4.1. Pravděpodobnostní



Obrázek 4.1: Maska popředí na základě dlouhodobého modelu pozadí

model byl prahovaný. Bílá barva reprezentuje oblasti, které s velkou pravděpodobností patří k pozadí scény. Černou barvou jsou znázorněny oblasti, které s velkou pravděpodobností patří k popředí scény.

Krátkodobý model pozadí

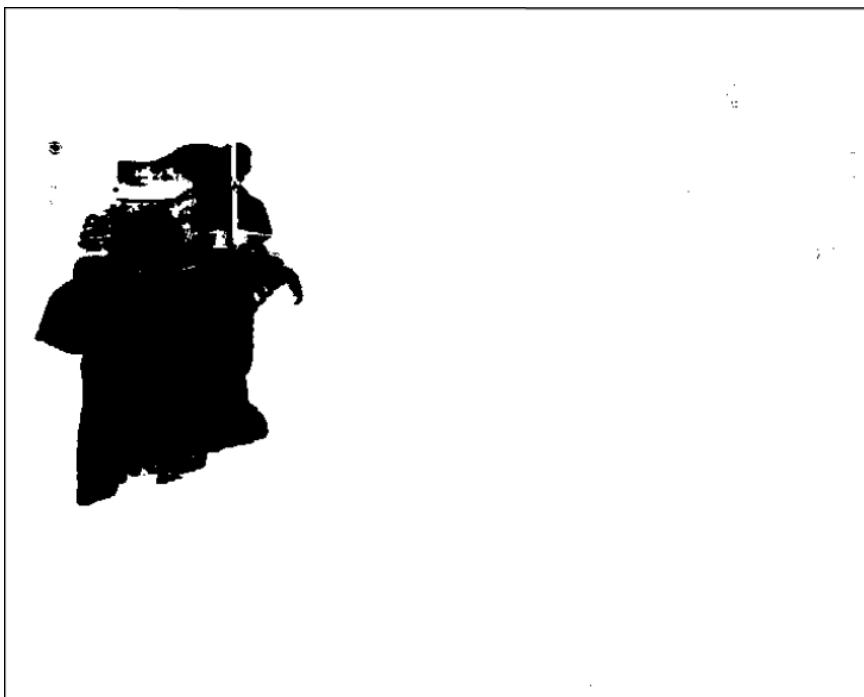
Krátkodobý model pozadí řeší stejný problém jako dlouhodobý model pozadí. Jedná se o odlišení objektů popředí od objektů pozadí ve videosekvenci, přičemž tyto objekty nejsou explicitně označeny pro vytvoření modelu pozadí. Přístup pro řešení tohoto problému je postavený na odlišném předpokladu, a to takovém, že jednotlivé objekty popředí vykazují výrazně větší změny polohy, než objekty, které jsou součástí pozadí, tudíž je možné je takto vyhledat.

Na obrázku 4.2 je znázorněný prahovaný model popředí získaný na základě krátkodobého modelu pozadí. Místa pravděpodobného výskytu objektu popředí jsou znázorněna černou barvou.

Další přístupy

Další přístupy segmentace mohou být založeny například na statistickém modelu barvy objektů popředí a detekci těchto barev v obraze[24], dále na metodách pro detekci hran, metodě watersheds a dalších.

Kvalitu výstupu jednotlivých metod lze ovlivnit kombinací výstupů těchto metod.



Obrázek 4.2: Maska popředí na základě krátkodobého modelu pozadí

4.2 Shluková analýza

Metody shlukové analýzy jsou z hlediska principu fungování komplementární k metodám segmentace. Zatímco metody segmentace se snaží rozdělit obraz, případně jiný typ prostoru, na samostatné regiony obsahující popis hledaných objektů, tak metody shlukové analýzy předpokládají, že hledaný objekt je v prohledávaném prostoru popsáný několikrát a snaží se tuto informaci z dat extrahovat. Shlukování lze definovat také jako dělení vzorů do tříd podle zadaného kritéria kvality.

Podrobnější informace o shlukové analýze lze nalézt v [53].

MeanShift

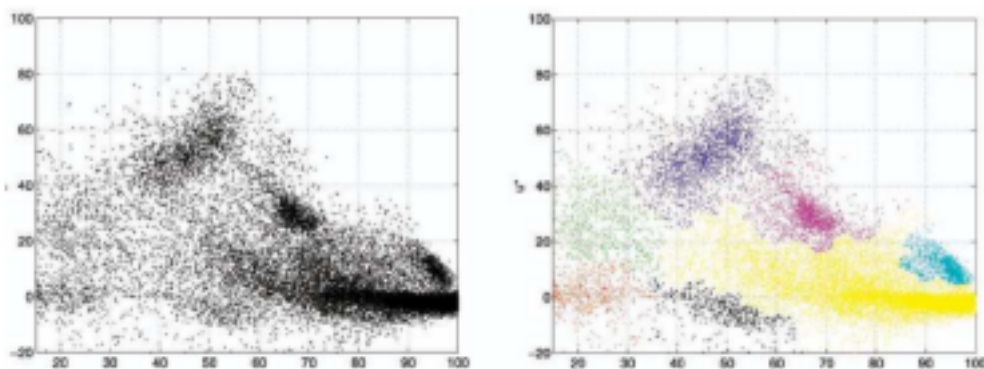
Algoritmus MeanShift je založený na předpokladu, že hustota bodů objektu se zvyšuje směrem ke středu objektu. Hustota rozložení bodů se vyhodnocuje lokálně podle vzorce 4.1.

$$f(x) = \frac{1}{Nh^d} \sum_{i=1}^N k \frac{\|x - x_i\|}{h} \quad (4.1)$$

Tento algoritmus je charakteristický tím, že je schopen od sebe rozlišit podlouhlé shluky, tak jako je to ilustrováno na obrázku 4.3. Podrobnější informace o principu tohoto algoritmu lze nalézt v [8], odkud byl rovněž převzatý obrázek.

k-Means

Metoda k-Means se česky nazývá metoda nejbližších středů. Kritériem pro přiřazení bodu do daného shluku je minimální vzdálenost od středu shluku podle rovnice 4.2.



Obrázek 4.3: Clustering metodou meanshift

$$H(y_1, \dots, y_N, \mu_1, \dots, \mu_N) = \sum_{i=1}^N \|x_i - \mu_{y_i}\|^2 \quad (4.2)$$

Funkci Algoritmu lze popsat následujícím schématem:

- do obrazu se vloží náhodně k bodů (středů jednotlivých shluků)
- u každého bodu se vypočítá vzdálenost od všech středů
- bod se přiřadí ke shluku, od kterého má nejmenší vzdálenost
- po zařazení všech bodů k některému ze shluků se vypočítá průměrná poloha všech bodů patřících k danému shluku a zvolí se jako nový střed daného shluku.
- postup se opakuje dokud se řešení neustálí

Algoritmus k -Means nedokáže zjistit kolik shluků se v obraze nachází a je potřeba tuto informaci explicitně definovat. Díky náhodné inicializaci může tento algoritmus dávat nestabilní výsledek, což znamená, že v případě jiné počáteční polohy středu shluku algoritmus vrátí jiný výsledek.

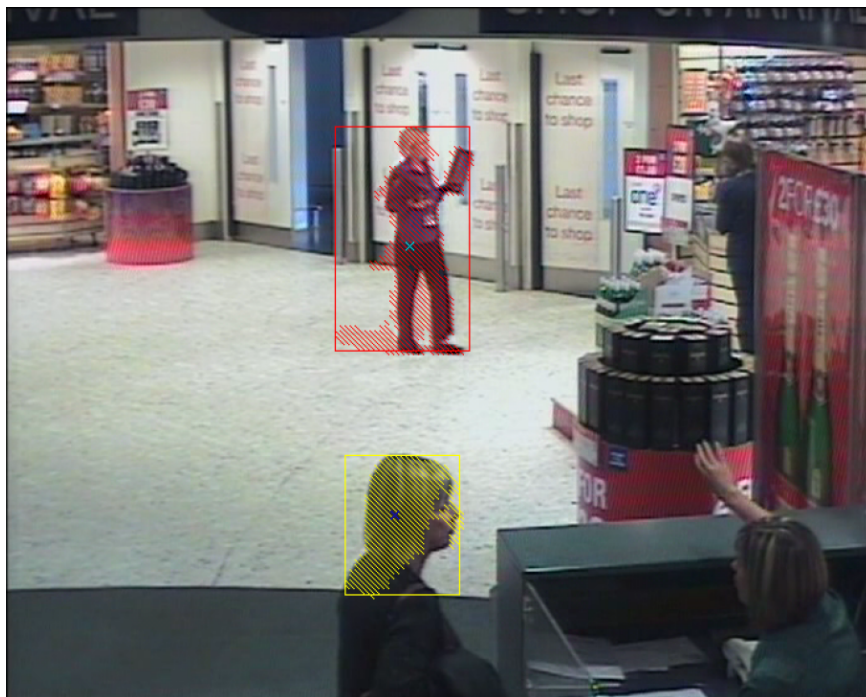
Connected Component Labeling

Metoda Connected Component Labeling, která je někdy nazývána také coloring nebo algoritmus semínkového vyplňování, patří mezi nejjednodušší metody shlukování. Je založena na vyplňování spojitých oblastí.

Spojitosť oblastí definujeme rekurzivně nad vzorkovaným obrazem tak, že pokud se ve dvou sousedních pixelech nachází objekt, tak je oblast spojitá. Sousednost pixelů můžeme definovat jako 4-okolí, 8-okolí nebo n -okolí.

Algoritmus lze implementovat rekurzivně podle definice nebo dvou průchodově, kdy se v prvním průchodu označí shluky podle zmenšeného okolí, které je definováno pixely, které už byly označeny. V druhém průchodu se přeznačkují oblasti, které patří ke stejnému shluku.

Na obrázku 4.4 je znázorněný výsledek shlukování metodou connected component labeling. Z původního obrazu byl metodou odečítání pozadí a prahováním vytvořený binární



Obrázek 4.4: Connected Component Labeling

obraz, který byl vyhlazen pomocí morfologických operací. V tomto binárním obrazu byly nalezeny shluky metodou connected component labeling. Tyto shluky byly různými barvami šrafovaně vyznačeny do původního obrazu. K jednotlivým shlukům byly vypočítány ohraňující obdélníky a průměrná poloha objektu, které byly do obrázku také zakresleny.

Ostatní metody

Pro shlukovou analýzu můžeme použít některé další metody.

Jednou z často používaných metod je *Fuzzy c-Means*, což je vlastně modifikace algoritmu *k-Means*. Úprava spočívá v přidání stupně příslušnosti daného bodu nebo subjektu shlukování k jednotlivým shlukům.

Další možností shlukování je pomocí *Gaussian Mixture Models*, což je metoda založená předpokladu, že jednotlivé objekty jsou ve shlucích rozmístěny s normálním rozložením pravděpodobnosti. Pro odhad rozložení pravděpodobnosti se používá například algoritmus *Expectation Maximization*.

4.3 Vyhledání klasifikací

Metody pro vyhledání objektů v obraze založené na klasifikaci přistupují k problému vyhledání objektů v obraze jiným způsobem než předchozí popisované přístupy. Základní myšlenkou těchto metod procházení obrazu pomocí okna a rozhodování jestli okno obsahuje nebo neobsahuje hledaný objekt.

Klasifikaci lze definovat také jako rozdělování vzorů do tříd podle kritéria definovaného učitelem „*supervised learning*“. Jednoduchý klasifikátor je schopen rozhodnout pouze problém příslušnosti k jedné třídě. Kombinací několika jednoduchých klasifikátorů lze intuitivně

vytvořit klasifikátor, rozhodující problém příslušnosti k 1-N třídám.

Základní schéma fungování těchto metod lze shrnout do těchto dvou kroků: fáze trénování a vyhodnocování. Ve fázi trénování se zvolí počáteční parametry metody, způsob a rychlost učení. Samotné učení spočívá v modifikaci parametrů klasifikátoru na základě pozitivních a negativních příkladů dané třídy.

Nejjednodušším příkladem takového klasifikátoru je korelace obrazu. Princip fungování této metody je vlastně stejný jako u konvoluční filtrace obrazu, přičemž jako jádro konvoluce se použije vzorový obraz. Takovýto filtr má největší odezvu v místech výskytu hledaného objektu.

K metodám klasifikace používaným pro identifikaci objektu patří například:

- Normalní Bayesovský Klasifikátor
- K nejbližších sousedů
- Support Vector Machines
- Rozhodovací stromy
- Boostování (například *Adaboost* nebo *Waldboost*)
- Náhodné stromy
- Neuronové sítě
- Skryté Markovovy Modely

Jednoduché klasifikátory se snaží odlišit vzory podle jejich vlastností. Tento přístup klasifikace obrazu přímo je velmi náchylný na geometrické transformace jako je například rotace nebo scale, apod. Pro zlepšení přesnosti klasifikace se jako vstup klasifikátoru místo obrazu používají příznaky obrazu. Jednotlivé typy příznaků jsou blíže popsány v kapitole 3.4.

V závislosti na použití se u klasifikátoru se po klasifikátorech požaduje schopnost zobecňovat, tedy například při klasifikaci obličejů se požaduje identifikace obličejů libovolného člověka a ne jenom skupiny lidí, na kterých byl klasifikátor natrénován.

Další informace o klasifikátorech lze nalézt v [53].

Kapitola 5

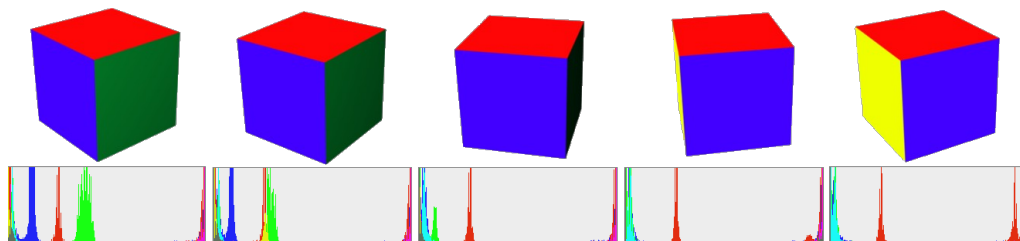
Počítačové vidění ve videosekvenci

Algoritmy popsané v předcházející kapitole se zabývají informacemi o objektech na jednom snímku videosekvence získané zpracováním toho snímku nebo části videosekvence. Tato kapitola se zabývá získáním ucelené informace o pohybu objektů ve videosekvenci. Cílem je tedy nalezení souvislosti v informacích o objektech ve dvou nebo více sousedních snímcích. Tuto činnost lze popsat jako snahu o spárování objektu nalezeném v jednom snímku s objekty nalezenými v následujícím snímku videosekvence. Na základě této souvislosti potom lze popsat polohu objektu ve videosekvenci pomocí její trajektorie. Trajektorii potom lze definovat jako polohu objektu v jednotlivých snímcích videosekvence.

Při snaze o spárování jednotlivých objektů v sousedních snímcích videosekvence dochází k řadě problémů. Algoritmus pro získání trajektorie objektu musí být odolný vůči vlivu perspektivy obrazu, deformacím obrazu vznikajícím díky optickým vlastnostem kamery, překrytí objektů a dalším jevům vznikajícím díky neúplným informacím o scéně.

Objekt pohybující se scénou v závislosti na vzdálenosti od kamery a parametrech optické soustavy kamery, mění svou velikost v obraze. Vliv perspektivní projekce na obraz lze popsat pomocí geometrických transformací a částečně kompenzovat. Dalším zdrojem problémů může být deformace obrazu vznikající na čočce videokamery. Způsob popisu těchto jevů a vypořádávání se s nimi je naznačeno v kapitole 3.3.

Při snímání scény pomocí videokamery dochází k tomu, že ve snímku není úplná informace pro popis všech objektů vyskytujících se ve scéně. Jednotlivé objekty jsou snímány pouze z jedné strany. V případě, že se objekt mezi jednotlivými snímky otočí podle osy X nebo osy Y, tak může výrazně změnit jeho vzhled. Rozdíl je ilustrován na obrázku 5.1. V horní části obrázku je z různých úhlů vykreslená krychle, která má na každé stěně odlišnou texturu. V dolní části obrázku je barevný histogram popisující texturu objektu pro každé natočení krychle.



Obrázek 5.1: Vliv rotace objektu na vzhled

Úhel snímání má podobný vliv i na tvar objektu. Toto lze intuitivně popsat příkladem

– tvar člověka popsany například jeho siluetou je v případě jeho snímání z anfasu¹ výrazně jiný než tvar siluety při snímání z profilu. Dalším výrazným problémem je, že i v případě, že by bylo možné modelovat tvar a texturu objektu třírozměrně, tak v reálných videosekvencích objekty mění svůj tvar. V případě pohybu člověka lze popsat tuto změnu tvaru jako změnu pózy. Příkladem tohoto jevu je změna pokrčení a natažení nohy při chůzi.

Nejsložitějším problémem je vzájemné překrytí objektů. V takovém případě je objekt na jednom ze snímků viditelný jenom částečně nebo není viditelný vůbec.

Principy jednotlivých metod popsanych v této kapitole byly čerpány z [5], [11] a [16].

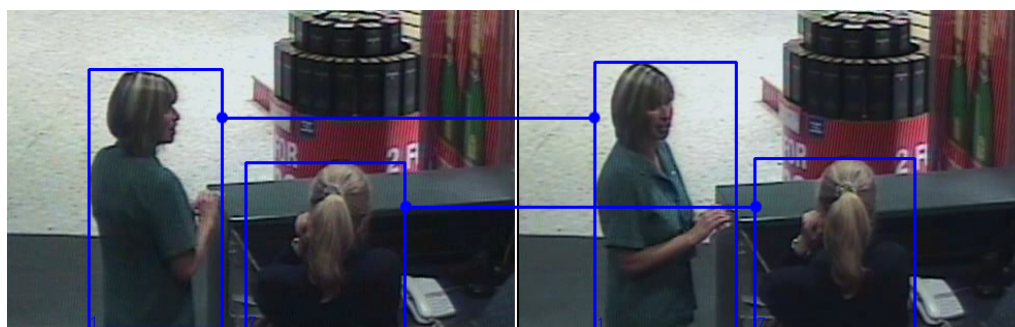
5.1 Spárování objektu ve dvou snímcích

Vyhledání a identifikace regionu v obraze korespondujícího s regionem v předcházejícím snímku nemusí být úplně jednoduchý úkol. Z videosekvence se vyberou dva sousední snímky, a v každém z nich se vyhledají oblasti, ve kterých se pravděpodobně nachází objekty zájmu. Pro každou z těchto oblastí se v sousedním snímku hledá oblast, která popisuje stejný objekt.

Spárování lze provést například pomocí shlukové analýzy, podobným způsobem jako pro statický obraz, což bylo podrobněji popsáno v kapitole 4.2. Pro zpracování lze použít přímo obraz rozšířený o časovou souřadnici, popis pomocí příznaků nebo nějakou zjednodušenou formu popisu objektů, například pomocí momentů. Další možností je použití klasifikačního přístupu (popsaného v kapitole 4.3), přičemž se klasifikátor trénuje na reprezentaci objektu patřící k jednomu ze snímků nebo více reprezentacích objektu z celé trajektorie. Pro zvolení jednoho z regionů lze použít porovnání odezvy jednotlivých klasifikátorů a vybrat maximální odezvu podle rovnice 5.1.

$$i^* = \arg \max_i P(c_i|O) \quad (5.1)$$

Příklad spárování dvou objektů ve dvou snímcích je ukázán na obrázku 5.2. V levé části obrázku je první snímek a v pravé části obrázku je druhý snímek. Poloha jednotlivých objektů je označena pomocí modrých obdélníků. Nalezená souvislost mezi objekty je naznačena čarou spojující jejich obdélníky. Pro zvýraznění rozdílů a zvýšení ilustrativnosti je vzájemná vzdálenost mezi snímky 5 snímků (tj. 0.2s, při frame rate 25 snímků za sekundu).



Obrázek 5.2: Spárování objektů v sousedních snímcích

V případě osoby vlevo je vidět výrazná změna jejího vzhledu ve 2D, přičemž tato změna tvaru a textury nepředstavuje pro lidské vnímání výrazný problém, ale při automatickém

¹zepředu (z franc. en face)

zpracování obrazu pomocí počítače to může být velký problém, protože pro určitý typ změn vzhledu objektu v obraze může být vhodnější popisovat tento objekt pomocí lokálních vzorů v obraze (např. Local Binary Patterns), pomocí barvy (např. barevného histogramu v HSV) nebo například pomocí tvaru celého objektu (popis pomocí b-spline křivek nebo aktivních kontur).

Algoritmy pro vyhledání objektu na základě popisu jeho vzhledu předpokládají, že je změna vzhledu objektu v jednotlivých snímcích videosekvence velmi malá nebo je objekt vůči okolní scéně dobře kontrastní, tudíž lze s vysokou pravděpodobností odhadnout jeho polohu.

MeanShift

Jedním z často používaných algoritmů pro sledování objektů je algoritmus MeanShift. Tento algoritmus využívá barevných histogramů objektu k jeho nalezení. Tento algoritmus hledá kandidátní oblasti, kde by se mohl vyskytovat sledovaný objekt, ze kterých následně vybírá nejlepší oblast ve smyslu Bhattacharyyaovy vzdálenosti. Hledaný objekt je reprezentovaný funkcí hustoty rozložení pravděpodobnosti v oblasti prostoru příznaků (barvou sledovaného objektu). Model sledovaného objektu se adaptivně upravuje v závislosti na míře shody objektu s kandidátními oblastí.

Metrika založená na Bhattacharyyaových koeficientech je definovaná na základě Euklidovské vzdálenosti porovnávaného vektoru příznaků kandidátního objektu a sledovaného objektu. Úloha výběru objektu z kandidátních řešení je definovaná jako minimalizace této vzdálenosti.

Výhodou tohoto algoritmu je schopnost sledovat objekty invariantně vůči natočení a poměrně dobrá odolnost vůči částečnému překrytí objektu. Další informace o tomto algoritmu je možné nalézt v [21] a [9].

5.2 Problém překrytí

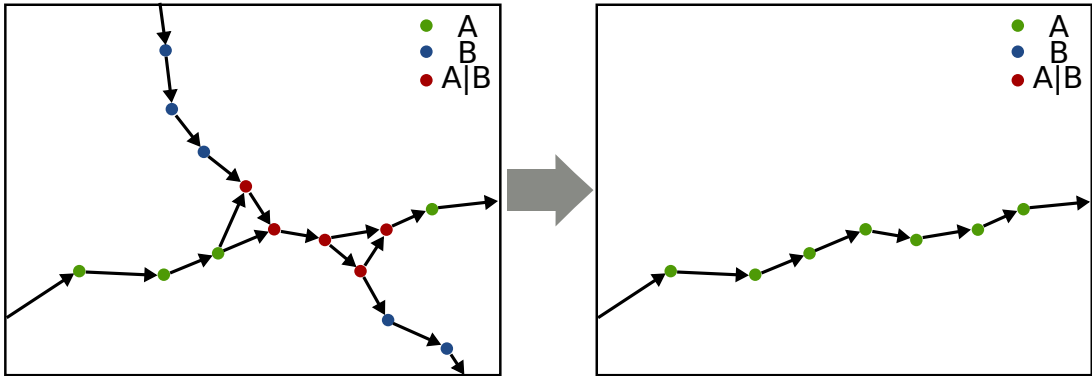
Ukazuje se, že největším problémem při sledování většího počtu objektů ve videosekvenci je problém překrývání se jednotlivých objektů. Překrytí objektu lze popsat tak, že jejich poloha je v rovině kamery vzájemně nerozlišitelná. Při sledování objektů v obraze může dojít ke třem případům, ve kterých není možné objekt sledovat. Prvním je částečné překrytí objektů, druhým je úplné překrytí objektů, posledním je opuštění scény.

V případě částečného překrytí je vhodné pokusit se objekty rozdělit a přesněji identifikovat jednotlivé objekty. Pro rozdělení objektů lze použít metody pro segmentaci obrazu popsané v kapitole 4.1. V případě úplného překrytí je nutné polohu objektu modelovat jiným způsobem, přičemž pokud objekt není vidět, tak je vhodné počítat i s variantou, že objekt opustil scénu.

Při sledování objektu v hromadné dopravě dochází k překrytí sledovaných objektů velmi často, u některých scénářů typických pro hromadnou dopravu lze dokonce předpokládat, že nebude možné objekt sledovat. Mezi takovéto scénáře patří například nastupování a vystupování cestujících v dopravní špičce. U těchto případech se předpokládá, že bude pravděpodobně vhodnější pro hledání bezpečnostních rizik, použít analýzu optického toku v textuře.

Grafová reprezentace

Jedním z možných řešení problému překrytí sledovaných objektů je pomocí grafové reprezentace [10]. Převod problému na grafovou reprezentaci $G = (U, H, c)$ spočívá v popsání jednotlivých objektů sledovaných v obraze jako uzlů U v orientovaném grafu, přičemž hrana $H = \{\{u, v\} : u, v \in U \wedge u \neq v\}$ reprezentuje rozpoznání objektu na jednotlivých snímcích. V případě překrytí jsou objekty pro každý snímek reprezentovány jedním uzlem. Přejechy mezi snímky jsou reprezentovány pomocí orientovaných hran, přičemž jejich ohodnocení c je dáno mírou shody jednotlivých oblastí neboli pravděpodobností, že oblasti v obraze reprezentují jeden objekt.



Obrázek 5.3: Extrakce trajektorie z grafu předpokládané polohy objektu

Na obrázku 5.3 je příklad grafové reprezentace problému. Na začátku videosekvence byly detekovány dva objekty A a B . V průběhu jejich pohybu došlo k úplnému překrytí těchto objektů tak, že nebylo možné navzájem rozlišit, což je popsáno v obrázku jako $A|B$. Po třech snímcích kdy tyto objekty nebylo možné navzájem rozlišit, se objekty rozdělily a na základě popisu jejich vzhledu došlo k jejich vzájemnému odlišení (objekty jsou označeny opět jako A a B). Pokud je překrytí objektů složitější, například tak, že se jeden objekt překrývá více objekty a není možné přesně určit jeho polohu v několika snímcích a existuje několik možných variant, tak se jeho trajektorie v sporných snímcích hledá jako nejkratší cesta grafem po cestě. Cesta v Grafu G mezi uzly u a v o délce n je definována jako sled 5.2 mezi uzly u, v takový, že platí vztah 5.3.

$$(u = w_0, h_1, w_1, h_2, \dots, w_{n-1}, h_n, w_n = v) \quad (5.2)$$

$$i \neq j \Rightarrow w_i \neq w_j \wedge h_i \neq h_j, 0 \leq i, j \leq n \quad (5.3)$$

Hledání nejkratší cesty v orientovaném grafu G s ohodnocením hran cenou c odpovídá hledání minimalizaci ceny průchodu grafem po všech cestách T o délce N tak, jak je to popsáno rovnicí 5.4 nebo v případě hledání nejpravděpodobnějšího výskytu objektu v dané oblasti obrazu hledání cesty s maximální pravděpodobností výskytu tohoto objektu, což je ekvivalentní problém.

$$j^* = \arg \min_j \sum_{i=0}^N (c_i | T_j) \quad (5.4)$$

Tento problém se lze řešit prohledáním všech možných cest z uzlu u do uzlu v podle Dijkstrova algoritmu pro hledání nejkratší cesty v orientovaném grafu nebo pomocí heuristických metod jako je například A^* . Informace o grafových strukturách byly čerpány hlavně z [50].

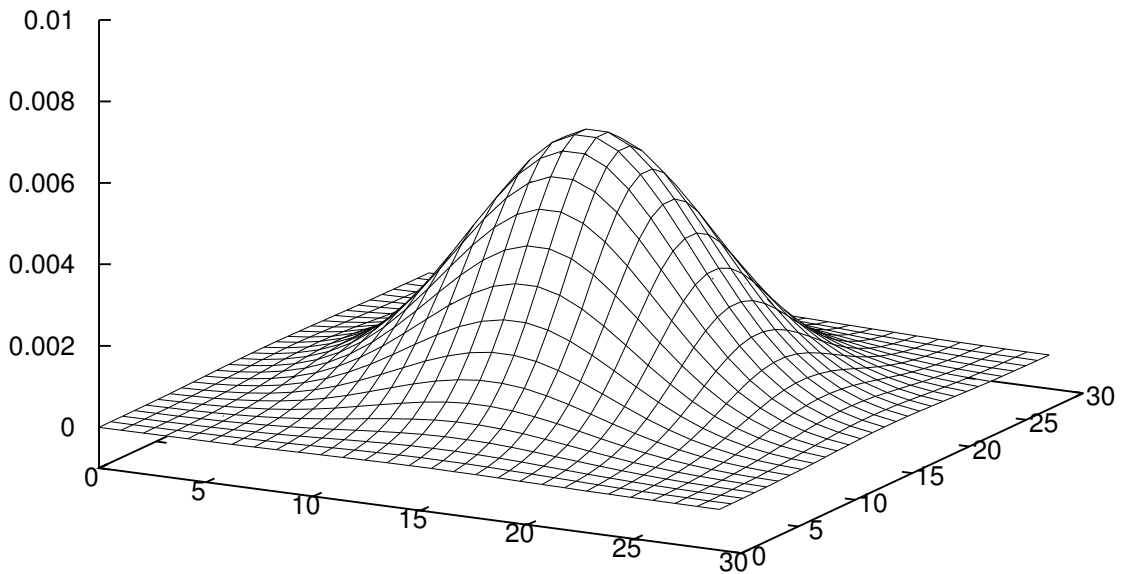
Kalmanův filtr

Při překrytí dvou objektů ve videosekvenci je snaha vyhledat polohu objektu pomocí vzhledu objektu. Videosekvence obsahuje navíc i temporální informaci o poloze objektu v předcházejících snímcích. Metoda používající Kalmanův filtr je založena na předpokladu, že pohyb objektu lze na základě předchozích měření poměrně přesně odhadnout.

Kalmanův filtr je algoritmus pro statistické modelování objektu a jeho pohybu. Princip fungování Kalmanova filtru je definovaný podle vzorce 5.5 [39]. Y_t ve vzorci je poloha bodu v čase t a Z_t změřená poloha bodu v čase t . Pro potřeby vyhodnocení algoritmu se definuje statický a dynamický model objektu.

$$p(Y_t|Z_{t-1}) = \int p(Y_t|Y_{t-1})p(Y_{t-1}|Z_{t-1})dY_{t-1} \quad (5.5)$$

Statický model popisuje samotný objekt – jak objekt vypadá a jakým způsobem je možné ho vyhledat v obraze. Možné způsoby reprezentace statického modelu objektu jsou rozebrány v kapitole 4. Pro účely Kalmanova filtru a modelování pohybu je možné objekt reprezentovat například bodem těžiště.



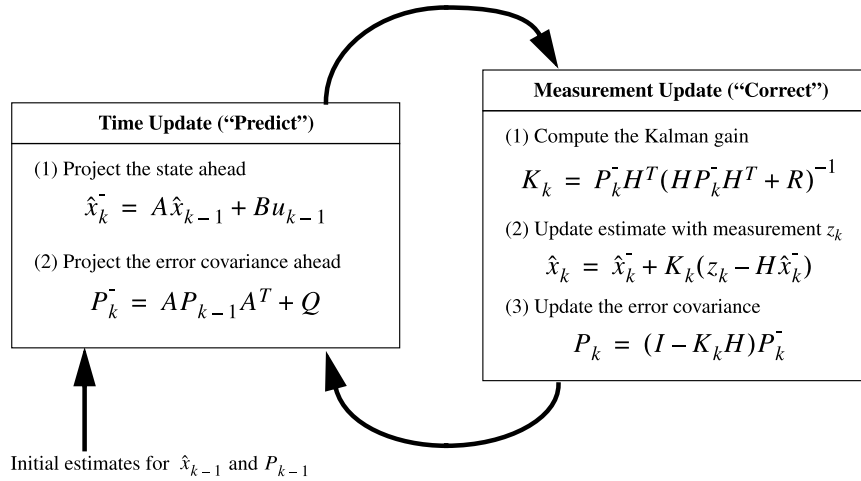
Obrázek 5.4: Normální rozložení pravděpodobnosti ve 2D

Dynamický model popisuje možnosti pohybu objektu. Odhad nové polohy objektu lze popsat pravděpodobností výskytu objektu ve všech bodech v obraze. V případě, že se objekt pohybuje po spojitě trajektorii lze jeho předpokládanou polohu modelovat jako normální rozložení pravděpodobnosti a popsat funkcí hustoty rozložení pravděpodobnosti popsanou vzorcem vzorcem 5.6, kde μ je střední hodnota a σ je rozptyl.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5.6)$$

Na obrázku 5.4 je znázorněno, jak se vzdáleností od střední hodnoty klesá pravděpodobnost výskytu objektu. Obrázek byl vytvořen pomocí programu gnuplot, pro vykreslení obrázku byla zvolena střední hodnota $\mu_{x,y} = [15, 15]$ a rozptyl $\sigma_{x,y} = [5, 5]$. V případě, že se Kalmanův filtr používá například k odhadu polohy robota, tak se pro odhad polohy používá také parametr „řízení robota“, který reflektuje změnu nastavení pohonných jednotek robota.

Princip fungování algoritmu lze názorně popsat pomocí obrázku 5.5 převzatého z [42]. Algoritmus se skládá ze dvou částí: predikce a korekce. Při spuštění algoritmu se vstup použije jako počáteční nastavení pro predikci objektu. Vypočítá se odhadovaná poloha objektu a odhad chyby, změní se nové souřadnice objektu, vypočítá se Kalmanův zisk a provede se korekce odhadu polohy objektu a korekce odhadu chyby měření polohy.



Obrázek 5.5: Schéma algoritmu Kalmanova filtru

Další informace týkající se Kalmanova filtru byly čerpány z [42] a [44].

Částicový filtr

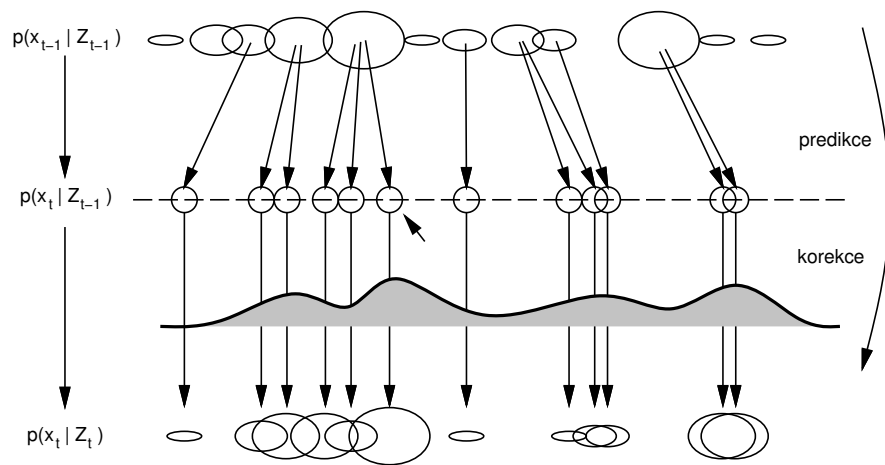
Částicové filtry jsou založené na stejném principu jako Kalmanův filtr. Hlavní rozdíl oproti Kalmanovu filtru je v dynamickém modelu objektu. Hustota pravděpodobnosti pohybu není popsána střední hodnotou a rozptylem, ale váhovanými částicemi reprezentujícími vzorkování dané funkce rozložení pravděpodobnosti.

Pro odhad modelu se obvykle používají Baysovské modely a sekvenční MCMC². Správně navržený částicový filtr může být mnohem rychlejší než MCMC². V porovnání s Rozšířeným Kalmanovým filtrem nebo Unscented Kalmanovým filtrem dokáže vytvořit optimální odhad modelu s minimálním počtem vzorků. Oba přístupy je možné kombinovat. Částicovým filtrům se někdy říká „sekvenční Monte Carlo metody“. Podobnost částicových filtrů s Kalmanovými filtry je diskutovaná v [45].

²Monte Carlo Markovské řetězcy

Monte Carlo je přístup k problému modelování pravděpodobné polohy objektu generováním částic, které aproximují hustotu rozložení pravděpodobnosti. V průběhu zpracování následujících snímku se poloha částic mění a upravuje se jejich váha v závislosti na snímání skutečné polohy objektu. Částicím, které lépe reprezentují skutečnou polohu objektu, se zvyšuje váha. Částicím, které hůře reprezentují částice se váha snižuje. Částice, které mají nulovou váhu se odstraňují. Aby nedošlo k postupnému odstranění všech částic, tak se provádí převzorkování.

Pro převzorkování se používá například Condensation algoritmus. Condensation je zkratka Conditional Density Propagation a v překladu znamená podmíněné předávání hustoty. Jedná se vlastně o to, že částicím s velkou váhou se sníží váha a v jejich okolí se generují nové částice. Princip převzorkování je ilustrován na obrázku 5.6 (obrázek je převzatý z [15]).



Obrázek 5.6: Mechanismus převzorkování: Condensation algoritmus

Existují další modifikace částicových filtrů. Tyto metody se většinou liší v algoritmu převzorkování.

- Auxiliary particle filter
- Gaussian particle filter
- Unscented particle filter
- Monte Carlo particle filter
- Gauss-Hermite particle filter
- Cost Reference particle filter

Další podrobnosti k problematice modelování pohybu pomocí částicových filtrů lze nalézt v [27] a v [15], ze kterého byl převzatý obrázek ilustrující převzorkování.

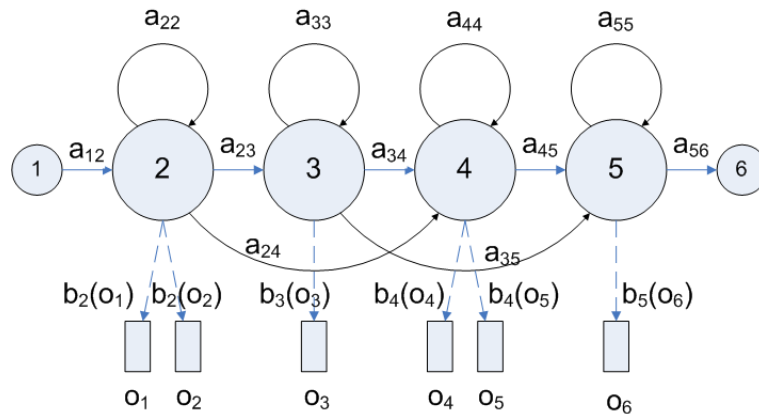
5.3 Analýza sekvenčních dat

Skryté Markovovy modely, dále jen HMM³, se řadí k metodám statistického rozpoznávání a strojového učení. Metody založené na HMM jsou vhodné pro klasifikaci sekvenčních dat o variabilní délce [5].

Za předpokladu, že lze najít vhodnou relaci mezi třídou trajektorie a třídou chování, tak lze aplikací HMM pro klasifikaci trajektorií identifikovat požadované chování objektů v obraze. Vstupem klasifikace je potenciálně nekonečná sekvence stavových vektorů $o(t) = [x, y, d_x, d_y, d_x^2, d_y^2]$ popisující polohu objektu v čase a její derivace (rychlost a zrychlení). Problém klasifikace trajektorie lze popsat jako identifikace třídy $c_i (i = 1..N)$, ke které trajektorie patří. Základní formulace tohoto problému je definovaná jako maximalizace podmíněné pravděpodobnosti:

$$i^* = \arg \max_i P(c_i|O) = \arg \max_i \frac{P(O|c_i)P(c_i)}{P(O)} \quad (5.7)$$

Protože není možné přímo zjistit $P(c_i|O)$, tak se používá Bayesův Teorém. Pro každou třídu se vytvoří samostatný model (viz. obrázek 5.7) reprezentující danou třídu. Tento model je konečný automat s K stavy generující sekvenci O na základě přechodových pravděpodobností modelu definovaných jako $a_{k,j}$. Kromě prvního a posledního stavu, model generuje výstupní hustoty pravděpodobnosti funkce $b_j(o(t))$, tak jak je to znázorněno na obrázku 5.7. Ten se natrénuje danou trénovací množinou dat, která se následně použije pro klasifikaci.



Obrázek 5.7: Skrytý Markovův model

Markovův model lze znázornit jako stavový automat, pro který platí, že jeho následující stav závisí pouze na aktuálním stavu. Skrytý Markovův model je vlastně jeden optimální průchod Markovovým modelem pro dané vstupní data. Pro nalezení skrytého modelu je nutné vyhodnotit všechny možné alternativy průchodu modelem. V průběhu vyhodnocování není možné s jistotou říci, který model je optimální pro daný vzorek.

V příkladu je definovaná přechodová matice $A = [a_{k,j}] (k, j = 1..K)$, která definuje pravděpodobnosti přechodu do následujícího stavu. Vstupní sekvence X může být dána například $X = 1, 2, 2, 3, 4, 4, 5, 6$. Pravděpodobnost průchodu sekvence X po cestě O modelem M je dána vztahem 5.8.

³Hidden Markov Models

$$P(O, X|M) = a_{x(o)x(1)} \prod_{t=1}^T b_{x(t)}(o_t) a_{x(t)x(t+1)} \quad (5.8)$$

Pro natrénování modelu se používá Baum-Welshův algoritmus [47]. Jedná se o zobecněný algoritmus Expectation Maximization a je definován vztahem 5.9, pomocí kterého se vypočítají nové přechodové pravděpodobnosti a statistické vlastnosti modelu.

$$P(O|M) = \sum_{\{X\}} P(O, X|M) \quad (5.9)$$

Pro vyhodnocení modelu a zjištění třídy, ke které trajektorie patří se používá Viterbiho algoritmus [47] definovaný rovnicí 5.10. Tento algoritmus vyhodnotí model maximalizací pravděpodobnosti toho, že trajektorii patří k danému modelu.

$$P^*(O|M) = \max_{\{X\}} P(O, X|M) \quad (5.10)$$

Další informace týkající se HMM je možné nalézt v [49], [47], [28] a [5]. Praktické implementace jednotlivých algoritmu lze nalézt v HTK toolitu [36] nebo STK toolkitu [35].

Kapitola 6

Analýza chování a stavu objektů

Prvním předpokladem pro úspěšnou komplexní analýzu chování a stavu objektu je to, že předchozí metody pro vyhledání a identifikaci objektu v obraze jsou schopny poskytnout dostatečně přesnou temporální informaci o stavu objektu. Stavem objektu jsou v tomto případě míněny jeho vlastnosti jako je například jeho poloha, vzhled nebo i další nevisuální vlastnosti. Cílem analýzy chování a stavu objektu je zjištění nejčastějších vzorů v chování, a predikce chování objektu a detekce abnormálního chování objektu.

Komplexní popis všech vlastností objektu všemi jeho parametry jako je například poloha objektu v čase nebo změna jeho textury, se jeví jako příliš složité a tudíž nevhodné pro další zpracování. Vhodnější než trajektorie objektu, trajektorie bodu na povrchu objektu nebo komplexnější popis změny polohy objektu, se jeví popis chování objektu na vyšší úrovni abstrakce. Zvolená úroveň abstrakce závisí na řešeném problému.

Intuitivně lze říci, že definování všech možných chování a stavů objektu, je velmi komplexní problém, jehož obecné řešení bude obtížně realizovatelné. Jednotlivé chování a stavy objektu se mohou navíc navzájem překrývat. Tento problém lze eliminovat definováním podmnožiny stavů a chování objektů, které jsou relevantní pro daný typ objektu a scény.

Slovním spojením „*možné překrývání chování a stavu objektu*“ se rozumí například situace „člověk utíká“, „člověk střílí z pistole“, přičemž toto chování člověka může nastat současně i samostatně. Pro detekci po sobě následujících chování je nutná buď schopnost rozlišit a oddělit jednotlivé chování navzájem nebo definovat podmnožinu všech chování, které mohou nastat jako celek. Bohužel neexistuje jednoznačný způsob jak dvě překrývající se chování jednoznačně rozdělit.

Obecně lze říci, že chování a stav objektu je závislý na chování ostatních objektů ve scéně. Vliv statických objektů na ostatní objekty v porovnání s vlivem jevů mimo snímanou scénu a vlivem chování a stavů dynamicky měnících se objektů lze považovat za konstantní. Tyto závislosti interakce objektů lze částečně eliminovat rozšířením scény přes více kamer, tak aby pokrývaly několikanásobně větší část světa a následně vyhodnocení lokálních interakcí, globálních interakcí a jejich vzájemných závislostí.

Dalším problémem, který je potřeba zohlednit při zobecnění metod analýzy objektů, je, že při získávání informací z kamery dochází ke zkreslení obrazu vlivem optické soustavy kamery, komprese, atd. Pro odstranění vlivu umístění kamery je možné na základě několika 2D souřadnic provést přepočítání do třírozměrného prostoru.

Problém analýzy chování a stavu objektu lze pojímat z několika různých pohledů. Intuitivně se jako správný typ dekompozice jeví rozdělení na chování nezávislé na interakci s ostatními dynamickými objekty a chování ovlivněnými interakcí s ostatními objekty.

Některé z výše zmíněných technik byly implementovány v rámci projektu Caretaker [27].

6.1 Ověření správnosti sledování objektů

Pro libovolné další zpracování je nutné ověřit jestli jsou data získané předcházejícími metodami správné. Toto vyhodnocení správnosti lze provést buď jednoduchým vykreslením získaných dat do videosekvence a vizuálním porovnáním nebo systematickým srovnáním výsledku zkoumané metody s požadovanými výsledky. V metodách počítačového vidění je snaha o napodobení lidského vnímání, za požadované výsledky lze tedy považovat výsledky získané na základě lidského vnímání.

V případě, že je zkoumaná metoda založena strojovém učení, tak dochází ke trénování dat a následné klasifikaci. Množina testovacích dat měla být disjunktní s množinou trénovacích dat. Tento postup se nazývá cross-validation. Při porovnávání výsledků s „Ground truth“¹ je možné rozlišovat pro každý vzorek dat míru shody výsledku metody s referenčním vzorkem nebo jenom binární reprezentaci této shody, tedy shoduje se nebo se neshoduje. Pro přesnější popis výsledku porovnání se rozlišují čtyři případy[39], které mohou nastat. Jejich popis je shrnutý v tabulce 6.1.

	Referenční	Testované
True Positive	1	1
False Positive	0	1
True Negative	0	0
False Negative	1	0

Tabulka 6.1: Taxonomie

Protože míra shody referenčních dat se zkoumanými daty není obvykle binární problém, ale výsledkem porovnání dvou vzorků je obvykle nějaké číslo popisující míru shody, tak se míra shody převádí pomocí nějakého prahu na binární reprezentaci. Tento práh je obvykle zadaný jako požadované veličina.

Porovnání zkoumané metody lze provést i s jinou metodou, výsledek je však potřeba interpretovat jako jejich vzájemnou rozdílnost, nikoliv jako správnost testované metody. Tento výsledek lze ho považovat za směrodatný pouze v případě, že referenční metoda je dostatečně robustní. V případě, že je možné u zkoumané metody nastavit nějaký parametr, který ovlivňuje její výsledek, tak je možné změnu odezvy zobrazit do grafu a následně zvolit hodnotu parametru podle požadovaných vlastností metody. Pro zobrazení úspěšnosti se používá nejčastěji graf závislosti False Positive Rate na True Positive Rate, tento graf se nazývá Receiver Operating Characteristic. Další často používanou vizualizací je P/R křivka, která zobrazuje závislost Precision na Recall, které jsou definovány v [39]. Při vyhodnocování úspěšnosti klasifikace se často používá stability based validation, která je založená na předpokladu, že metoda by měla dávat stejné výsledky v případě, že pro trénování se použijí dvě rozdílné množiny dat.

Příkladem adhoc stanovené metriky pro ověření správnosti výstupu předcházejících metod může být předpoklad, že sledované objekty pro video získané z kamery umístěné na jednom konkrétním místě mají maximální velikost 600px a minimální velikost 100px. Podle této adhoc stanovené metriky lze detekovat poměrně snadno výrazné zhoršení výstupů metody pro získávání informací o poloze a velikosti objektu. Dalším příkladem adhoc omezení pro scénu může být stanovení míst, ve kterých objekt může opustit scénu.

¹obecně používaný pojem, který označuje referenční množinu dat

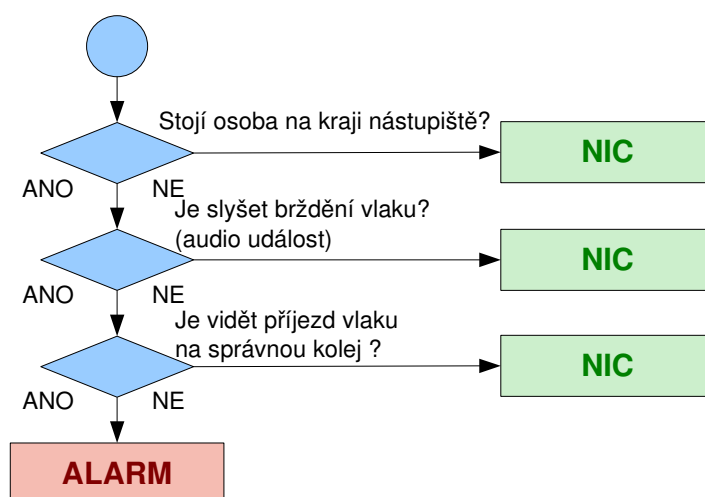
Tímto způsobem lze ověřovat různé kroky metod pro sledování objektů a relativně nezávisle vyhodnocovat jejich úspěšnost a ověřovat různé předpoklady, jako například předpoklad, že pro vyhledání objektu ve scéně nestačí informace o poloze objektu v jednom snímku, ale je potřeba složitější informace z části videosekvence. Dalším příkladem může být krok zpracování videosekvence zabývající se řešením kolizí objektů ve scéně. Zkoumanou informací je tedy počet řešených kolizí a počet úspěšně vyřešených kolizí.

6.2 Kombinované události

Metoda kombinovaných událostí vychází z předpokladu, že chování objektu lze jednoznačně rozpoznat z dílčích diskrétních událostí. Jednotlivé komplexní chování, jehož detekce je požadována v dané scéně, je definována jako několik dílčích jednoduchých chování nebo stavů objektů. Toto dílčí chování lze popsat pomocí symbolů výrokové logiky. Pomocí formulí výrokové logiky pak lze popsat „vzory hledaného chování“.

Dílčí chování (termy výrokové logiky) lze definovat jako jednoduché chování nebo stav týkající se jednoho nebo několika objektů. Tímto přístupem je tedy možné postihnout interakci několika objektů. V praktické implementaci potom tato metoda pracuje nad určitým časovým oknem, a proto je nutné rozšířit popis problému o tuto informaci pomocí temporální logiky.

Princip této metody je ilustrován na obrázku 6.1.



Obrázek 6.1: Detekce komplexního chování

Dalšími vstupními daty pro tuto metodu mohou být například:

- Souřadnice objektu (objekt se nachází v zakázané oblasti)
- Globální informace (denní doba, událost odjezd poslední noční linky)
- Vlastnost objektu (objekt má tvar a velikost zavazadla)
- Stav objektu (opuštěné zavazadlo je objekt, který se dlouhou dobu nehýbe)

Bližší informace o této metodě jsou uvedeny v [27].

Kapitola 7

Implementace

Tato kapitola se zabývá implementací některých algoritmů zmíněných v předcházejících kapitolách, způsobem jejich zabudování do programových systémů, zhodnocením jejich výsledků a možnostmi jejich dalšího vývoje.

Protože zvolený přístup rozděluje zpracování videosekvence do bloků (viz. kapitola 2), tak je vhodné dekomponovat problém začlenění do širších programových systémů pomocí rozhraní mezi jednotlivými funkčními bloky. Toto rozhraní umožní zabudování vybraného bloku nebo všech bloků do jiného programového systému.

Při implementaci jednotlivých komponent bylo použito programovacích jazyků *C* a *C++*. Dále byla použita knihovna OpenCV[13], která poskytuje jednoduchý přístup k obrazu uloženém ve videosekvenci nebo obrazu aktuálně snímaném videokamerou. Tato knihovna také obsahuje velké množství algoritmů pro zpracování obrazu, počítačové vidění a strojové učení. Pro modelování pohybu objektů a jejich následnou analýzu byla mimo jiné použita knihovna HTK[36]. Na propojení některých komponent navzájem nebo s testovacími programy bylo použito skriptovacích jazyků *PHP* a *Bash*.

7.1 Integrace do programových systémů

Způsob předávání dat je velmi komplexní problém, který se řeší v mnoha systémech. Definování abstraktního rozhraní umožňuje snadno zvyšovat komplexnost programu a nezávisle na tom budovat robustní a snadno zaměnitelné algoritmy pro řešení jednotlivých dílčích problémů.

Rozhraní pro předávání dat je možné definovat na úrovni zdrojových kódů nebo na úrovni dat. Existuje velké množství standardů, které se zabývají touto problematikou. Tato podkapitola se zabývá některými z nich vhodnými pro předávání audiovizuálních dat nebo sémantických dat, která je popisují.

Kompatibilita zdrojových kódů

Kompatibilita na úrovni zdrojových kódů je dána API¹. Umožňuje překlad a používání modulů programu bez znalosti implementace těchto modulů. V závislosti od použitého programovacího jazyka se odvíjí způsob předávání dat na úrovni programu. Programovací jazyky umožňující tento způsob předávání dat jsou obvykle založeny na modulárním paradigmatu[29].

¹Application Programming Interface

Na úrovni jazyka C nebo C++ je možné vytvářet programové rozhraní pomocí dynamických nebo statických knihoven. Začlenění do širšího projektu se skládá z přidání hlavičkových souborů a úpravy linkování² programu. Při statickém linkování se kód algoritmu vkládá přímo do výsledného programu, kdežto při dynamickém linkování je do programu vloženo pouze volání funkce, jejíž program je definovaný v dynamické knihovně.

Rozhraní na úrovni zdrojových kódů k metodám pro sledování objektů je popsáno v příloze A. Návrh jednotlivých komponent systému pro sledování objektů je detailněji popsán pomocí ER diagramu v příloze B.

Datová kompatibilita rozhraní

Rozhraní na úrovni předávání dat lze definovat buď binárně nebo textově. Kompatibilita programů na úrovni dat je často velký problém a existuje snaha o jejich standardizaci na úrovni podnikových norem nebo ISO norem. Způsobem uchovávání audiovizuálních dat se zabývá norma MPEG[14], pro uchování multimediálních dat existuje standard MPEG-7[39]. Pro předávání dat v textové podobě existuje také standardizovaný protokol SOAP[33], který používá pro ukládání dat formát XML.

Pro předávání dat na audiovizuálních dat v binární podobě lze použít protokol RTP³.

Textový datový formát XML umožňuje přizpůsobení tohoto formátu pro konkrétní aplikaci včetně možnosti kontroly kompatibility formátu a vytváření různých verzí. Hlavní výhodou je možnost ruční kontroly obsahu souboru. Jednou z možností definice konkrétní modifikace XML je pomocí Relax NG schématu[37]. Toto schéma lze převést do XML schématu, ale na rozdíl od něj je lépe čitelné pro člověka.

Rozhraní vytvořené na datové úrovni umožňuje velmi vysokou kompatibilitu napříč různými programovacími jazyky a operačními systémy. Výhodou XML popisu dat je, že kromě samotných dat obsahuje i metadata popisující význam těchto dat, tudíž je často možné je požit bez jejich dalšího přesného popisu. Při použití protokolu SOAP pro předávání dat je navíc možné relativně snadno realizovat zpracování těchto dat na více počítačích.

Pro potřebu předávání dat o sledovaných objektech na úrovni dat bylo navrženo rozhraní ve formátu XML. Tento formát byl popsán pomocí Relax NG schématu. Toto schéma lze nalézt v příloze A.

7.2 Vyhledání objektů v obraze

Pro testování a vyhodnocení algoritmů pro sledování objektů v obraze byly použity videosekvence pořízené ve stanicích Římského a Turínského metra[41] a na letišti Gatwick[38]. Videokamery, kterými byla pořízena testovací data, jsou umístěny staticky, proto pro zpracování záznamů získaných z těchto kamer lze s výhodou použít algoritmy založené na odečítání pozadí popsané v kapitole 4.1. Jednotlivé objekty jsou potom popsány pomocí pravděpodobnosti jejich výskytu v daném snímku obrazu a modelem jejich vzhledu.

Pro prvotní vyhledání potenciálně zajímavých objektů za účelem jejich sledování byla použita metoda odečítání pozadí založená na dlouhodobém modelu pozadí. Protože tato metoda byla relativně hodně citlivá šum videokamery a změny osvětlení, tak byl tento pravděpodobnostní model výskytu objektu doplněn o krátkodobý model pozadí[1]. Při použití krátkodobého modelu pozadí je nutné explicitně ošetřit případy zastavení objektu jako je například „odložené zavazadlo“.

²sestavení programu, součást překladu

³Real-time Transport Protocol

Toto prvotní vyhledání objektu bylo rozšířeno o vyhledání objektu na základě jeho modelu vzhledu. Tento „model vzhledu“ je vlastně definován jako abstraktní popis objektu pomocí, kterého lze objekt vyhledat v obraze. Tento model může být definován jako jednoúčelový klasifikátor. Tento abstraktní přístup umožňuje vytvoření více různých modelů vzhledu objektu založených na různých metodách, případně kombinování několika různých způsobů popisů objektu. Jako referenční přístup pro popis modelu vzhledu objektu byl použitý jeho barevný histogram.

Možnosti vylepšení komponenty pro detekci objektu je mnoho. První z nich je zvýšení robustnosti prvotního vyhledání objektu na základě modelu pozadí, zvýšení jeho odolnosti vůči šumu, případně úpravu této metody tak, aby byla použitelná v případě pohyblivé kamery.

Vhodným místem pro zlepšení je také detektor polohy známého objektu, případně porovnání úspěšnosti detekce objektu ve scéně pomocí různých detektorů nebo pomocí jejich kombinace. Nabízí se použití vyhledání pomocí klasifikace nebo pomocí shlukové analýzy. Pro tyto metody je vhodné obraz převést do prostoru příznaků. Tyto příznaky potom popisují tvar nebo texturu objektu.

Jednou z možností jak je možné popsat tvar objektu v obraze je například pomocí kontury objektu. Samotné vyhledání by potom mohlo být založené na tzv. „aktivních konturách“, přičemž kontura objektu nalezeného ve snímku by byla založena na předpokládané kontuře objektu.

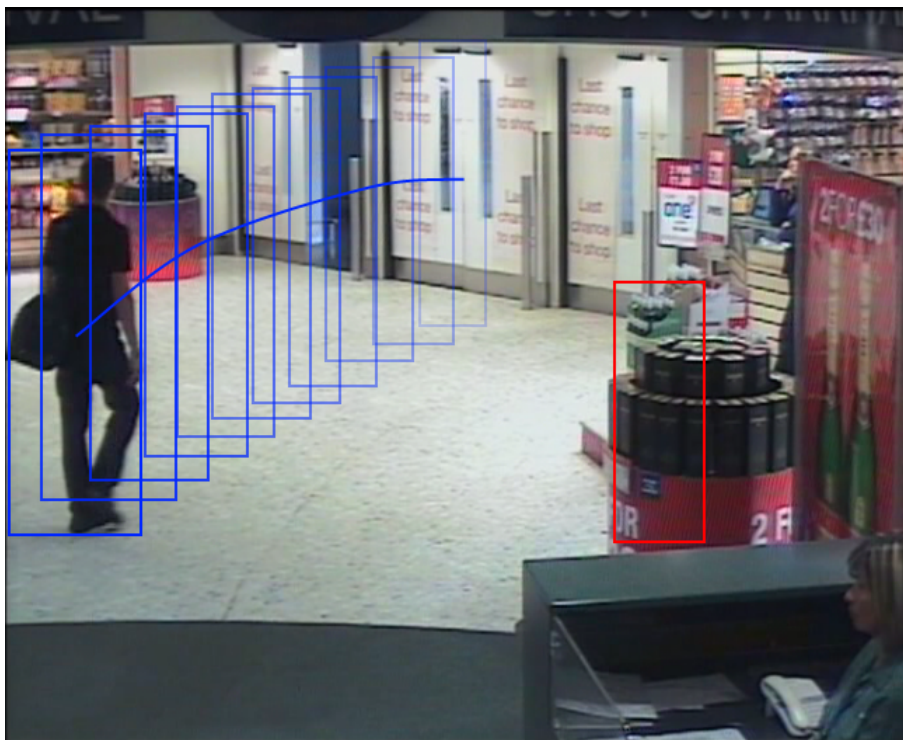
7.3 Sledování objektů v obraze

Vyhledání všech objektů v jednom snímku videosekvence je obtížný problém a jeho úspěšnost je hodně závislá na použité metodě. V případě, že je vyhledání objektu založeno na detektoru využívajícím krátkodobý model pozadí, tak je možné vyhledat pouze pohybující se objekty. V případě, že jedním z požadavků je vyhledávání opuštěných zavazadel, tak je tato detekce nedostatečná. Dalším problémem, který je potřeba řešit, je vzájemné překrytí objektů.

Jednou z otázek je, jakým způsobem a kdy oddělit jednotlivé objekty navzájem. Intuitivní přístup k tomuto problému směřuje ke srovnání s lidským vnímáním. Pomocí příkladu lze tento problém popsat například jako otázku jestli „člověk se zavazadlem“ má být popsán jako dva objekty „člověk“ a „zavazadlo“ nebo jako jeden objekt. Tento problém byl nakonec řešen tak, že k rozdělení jednotlivých objektů dochází jenom v případě, že jsou detekovány odděleně.

Pro řešení těchto problémů byla přidána k informaci o poloze objektu z jednoho snímku navíc i temporální informace. V případě detekce objektu v jednom snímku je nežádoucí, aby výstupem metody pro vyhledání objektu v obraze byly falešné detekce objektu, při aplikaci metody pro vyhledání objektu na videosekvenci je naopak žádoucí nastavit metodu tak, aby vyhledala objekt ve 100% případech. Falešné detekce objektu se následně odstraní na základě modelu pohybu objektu. Model pohybu objektu byl implementovaný pomocí částicového filtru. Na obrázku 7.1 je naznačena detekce objektu v několika snímcích videosekvence. Detekce postavy procházející scénou je znázorněna modrými obdélníky, falešná detekce osoby červeným obdélníkem.

Implementovaný algoritmus pro sledování objektů lze popsat následovně: (1) Sledování objektu je inicializováno metodou pro vyhledání objektu na základě modelu pozadí. (2) Následně se vytvoří model vzhledu a model pohybu objektu. Model vzhledu je založený na tvaru a barvě objektu. Tvar je popsán maskou objektu a barva barevným histogramem.



Obrázek 7.1: Detekce objektu ve více snímcích

(3) Na základě modelu pohybu a vzhledu se vytvoří několik hypotéz o možné poloze objektu. (4) Tyto hypotézy se vyhodnotí na jejich základě se opraví model vzhledu a pohybu objektu. Jejich vyhodnocení se provede i vzhledem k dlouhodobému modelu pozadí. (5) Postupně se eliminuje vliv objektů na model popředí, přičemž v případě, že se ve stejné oblasti vyskytuje i jiný objekt, tak se provede opětovné vyhodnocení hypotéz o poloze tohoto objektu a model jeho vzhledu se neupravuje. (6) V případě, že hypotéza o poloze objektu nepokryvá souvislou oblast a objekt se nepřekrývá s jiným objektem, dojde k převzorkování a rozdělení na několik objektů.

Při částečném překrytí objektů se implementovaný algoritmus nejdříve snaží podle modelu vzhledu objektu určit kde se nachází viditelná část objektu. Následně se snaží eliminovat vliv překrytí na vzhled ostatních skrytých objektů.

Vyhodnocení úspěšnosti

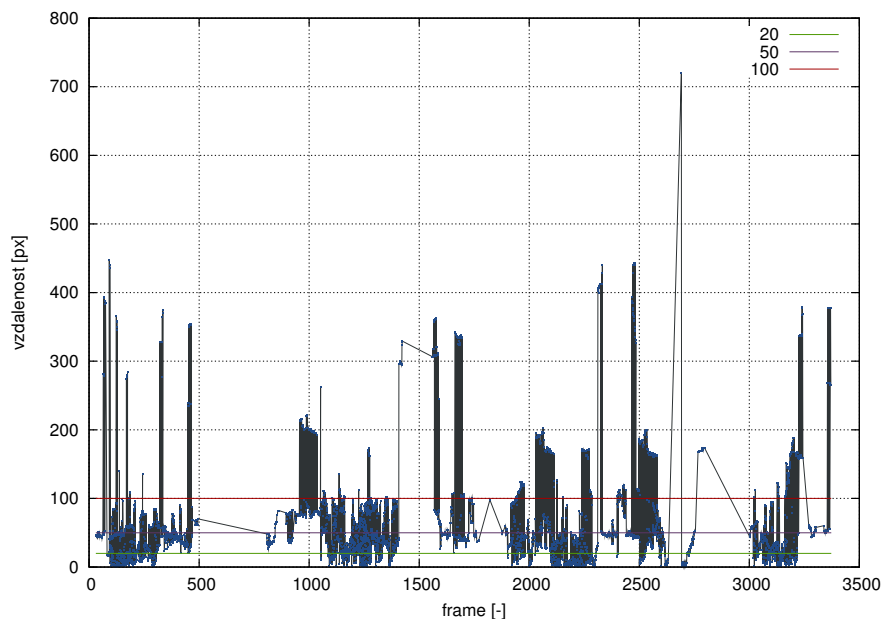
Vyhodnocení úspěšnosti metod pro sledování objektů je nedílnou součástí popisu metody. Aby bylo možné popsat úspěšnost je nutné porovnat informace o poloze objektů v čase s správnou informací⁴. Tato anotace objektu obsahuje popis očekávaného výsledku. Je možné ji vytvořit ručně, automaticky pomocí nějaké jiné spolehlivé metody pro sledování objektů, nebo poloautomaticky. Poloautomatický přístup může kombinovat korekci automatické detekce nebo interpolaci částečné ruční anotace. Pro účely vyhodnocení metod pro sledování objektu byl vytvořen program pro anotaci trajektorie a s jeho pomocí byl vytvořen popis polohy objektů ve videosekvenci. Tento popis zahrnoval pouze popis pohybu objektů spojených s pohybem lidí (tedy například člověk se zavazadlem, ale ne pohyb dveří).

⁴ground-truth

Prvním krokem vyhodnocení je zobrazení výsledků sledování a jejich vizuální ověření. Tento přístup sice neposkytuje přesné numerické údaje o výsledku metody, ale je poskytuje poměrně velké množství informace, která většinou postačuje k počátečnímu nastavení metody.

Pro vyhodnocení úspěšnosti bylo anotováno 3 500 snímků ze záznamu pořízeného pomocí kamery číslo 1 na letišti Gatwick[38]. Ve zmíněné videosekvenci bylo označeno 18 objektů celkem 11 563 krát. Pro porovnání úspěšnosti vyhodnocení zkoumána úvaha, že sledování objektů je stejná úloha jako úloha vyhledání objektů v obraze s tím, že navíc byla přidána informace o vzájemné korespondenci detekce v sousedních snímcích videosekvence a bylo vyřešeno překrytí objektů. Na základě této úvahy lze předpokládat, že vzdálenost anotovaného objektu od nejbližšího objektu detekovaném v jednom snímku bude v ideálním případě stejná jako vzdálenost anotovaného objektu od objektu nalezeného pomocí sledování. Tato poloha se bude lišit pouze v případě detekce skupiny objektů jako jednoho objektu a chybného sledování objektu.

Z tohoto důvodu byl vytvořen graf 7.2 znázorňující tuto vzdálenost. Pro anotovanou videosekvenci bylo nalezeno 4887 objektů (proti 11 561 anotovaným objektům). V grafu jsou vyznačeny adhoc zvolené mezní hodnoty, které by mohli být kritériem úspěšného vyhledání objektu. Detailnější přehled počtu úspěšného nalezení objektu je znázorněn v tabulce 7.2. Podle grafu lze toto pozorování rozdělit do několika skupin: úspěšné vyhledání objektu (do 20px), vyhledání objektu v těsné blízkosti (do 100px), což indikuje spojení několika objektů do jedné oblasti a chybné vyhledání objektu (nad 100px).



Obrázek 7.2: Vzdálenost nejbližšího objektu od anotace

Pro ohodnocení úspěšnosti řešení problému překrytí je také relativně důležitý údaj o počtu řešených částečných a celkových překrytí. Míra překrytí objektů se poměrně špatně definuje, pro účely vyhodnocení byly u jednotlivých objektů popsány tři případy: objekt je viditelný, překrytí je objektu malé a překrytí objektu je velké. Přičemž velké překrytí bylo definováno jako zakrytí více než 50 % plochy povrchu objektu viditelného v rovině kamery jiným objektem popředí v daném snímku videosekvence. V testovací videosekvenci

vzdálenost	20px	50px	100px
počet výskytů [obj.]	1028	2846	4069
počet výskytů [%]	21.03	58.23	83.27

Tabulka 7.1: Vzdálenost nejbližšího objektu od anotace

bylo anotováno 515 menších překrytí a 125 velkých překrytí. Překrytí objektu lze vyhodnotit také pomocí vzdálenosti anotace od objektu vyhledaného pomocí sledování objektů v porovnání s vyhledáním bez sledování objektu.

Jedním z ukazatelů správnosti metody může být závislost přesnosti měření na míře překrytí objektů. Dalším zajímavým údajem může být sledování změny chyby odhadu polohy objektu s dobou překrytí objektů.

Pomocí jednoduchého vizuálního porovnání bylo zjištěno, že implementovaná metody pravděpodobně není optimálně nastavena a poskytuje horší výsledky než jednoduché vyhledání objektu. Než se podaří upravit nastavení metody tak, aby dávala relevantní výsledky znatelné při vizuálním porovnání, tak není nutné zpracovávat detailnější popis chybovosti metody.

Možnosti dalšího vývoje

Při řešení problému bylo potřeba zajistit vytvoření anotací, nastavení parametrů detekce a vyhodnocení úspěšnosti metody pro vyhledání a sledování objektu. Zlepšení výsledků detekce lze dosáhnout použitím robustnějších metod pro detekci dílčích jevů ve videosekvenci nebo jejich lepším nastavením a lepším vzájemným kombinováním výsledků jednotlivých metod pro vyhledání objektu.

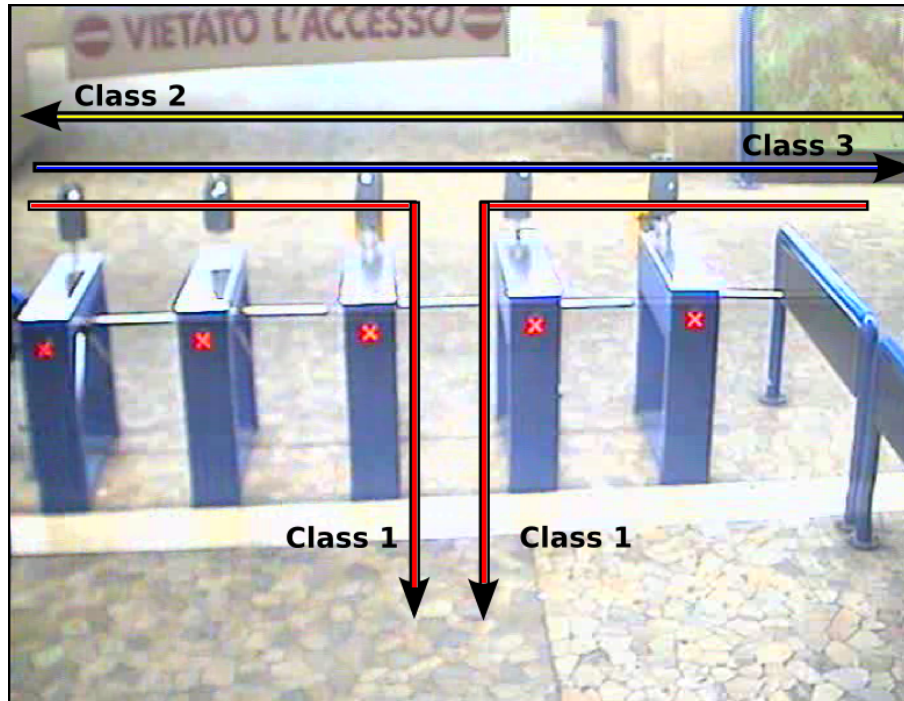
Problém optimálního nastavení metody je potřeba řešit při změně podmínek detekce. Tyto podmínky se mohou pro různé scény velmi výrazně lišit, proto je jedním z nejzajímavějších směrů dalšího vývoje programu současný vývoj algoritmů pro automatické vyhodnocení úspěšnosti detekce a následné nastavení parametrů pro ni potřebných.

Největším problémem pro kvalitní vyhodnocení výsledků je existence kvalitních anotací dat. Pro popis očekávané polohy 18 objektů ve videosekvenci je potřeba popsat cca 11 000 poloh v jednotlivých snímcích. Pro kvalitní vyhodnocení úspěšnosti by vhodné ověřovat úspěšnost algoritmu v řádově na tisících objektů v různých scénách. Pro vytvoření takového množství anotací by bylo potřeba několika stovek tisíc hodin práce. Tuto činnost je možné zefektivnit za pomoci vhodných nástrojů, kterými stačí anotovat pouze jeden ze sta snímků a ostatní se nějakým způsobem interpolují. Ideální způsob pro vytvoření velkého množství takto velkého množství dat jsou kolaborativní nástroje, které umožní jejich masivně paralelní zpracování.

7.4 Analýza chování objektů

Informace o poloze objektů v čase získaná metodami pro sledování objektů ze záznamů získané od partnerů v projektu Caretaker [41] byla analyzována pomocí HMM. V prvním kroku byla zvolena taková scéna, aby statické objekty pokud možno minimálně ovlivňovaly chování dynamických objektů. Zvolená scéna je znázorněna na obrázku 7.3. V dané scéně byli určeny všechny normální trajektorie (vyznačené v obrázku) a vytvořena trénovací sada

dat (pomocí [3]), tak že se každá trajektorie, u které to bylo možné, přiřadila do jedné ze tříd.



Obrázek 7.3: Všechny normální trajektorie pro scénu

Trajektorie objektů, které nebylo možné při anotaci zařadit do žádné ze tříd kvůli vstupnímu šumu byly anotované jako samostatná třída.

Nad anotovanými daty byly následně provedeny dva experimenty. Data popisovala trajektorii objektu ve videosekvenci jako sekvenci čtveřic $[x, y, d_x, d_y]$ a byla dodána od partnera v projektu Caretaker[41]. Délka těchto sekvencí se zahrnovala pohyb objektů většinou v rozmezí od 10 do 400 snímků videosekvence. První experiment byl proveden nad filtrovanými daty a do procesu trénování ani vyhodnocení nebyly zahrnuty data obsahující šum. Přesnost klasifikace u filtrovaných dat dosahovala 91.92 %, podrobnější přehled o úspěšnosti klasifikace pro jednotlivé třídy jsou shrnuty v tabulce 7.2. Pro trénování a testování bylo použito cca 800 trajektorií. Hlavním měřítkem pro zařazení trajektorie do třídy byla nejvyšší pravděpodobnost příslušnosti k dané třídě.

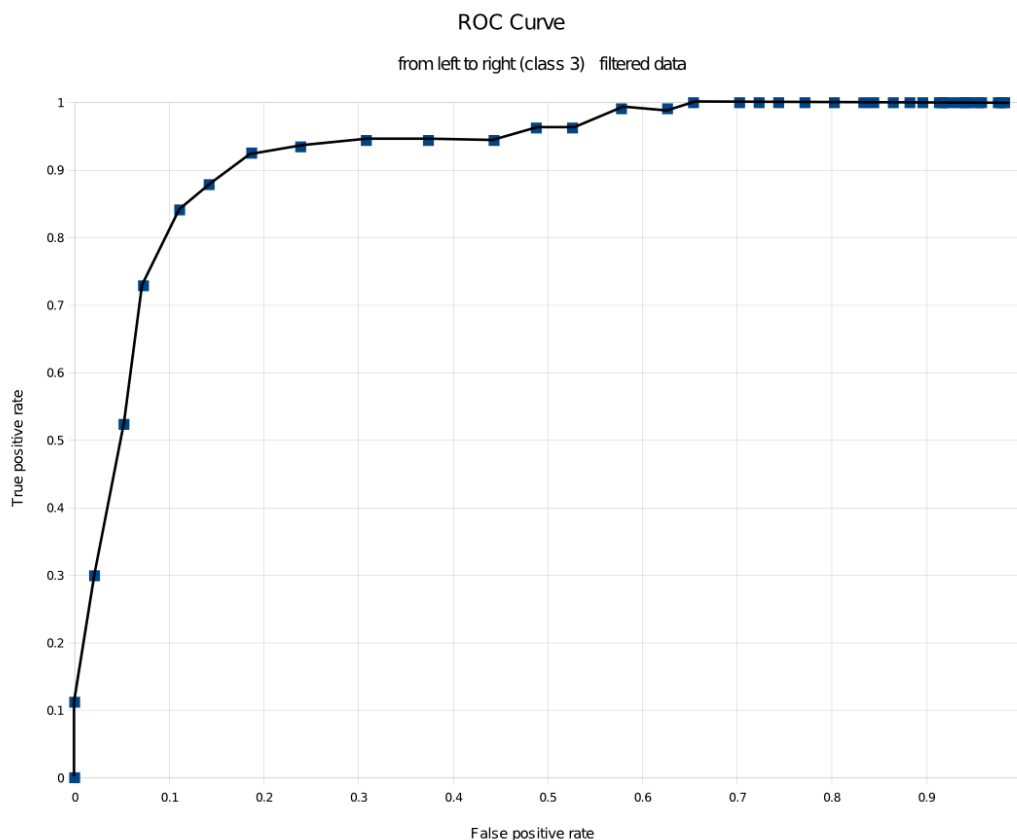
class	TP [%]	TN [%]	FP [%]	FN [%]
1	62.12	32.83	0.51	4.55
2	5.30	87.37	6.31	1.01
3	24.49	72.22	0.76	2.53

Tabulka 7.2: Výsledky klasifikace trajektorií na filtrovaných datech.

Pro třídu chování objektů přesouvajících se z levé části scény do pravé části scény popsanou příslušným HMM byla vytvořena ROC⁵ křivka (viz. obrázek 7.4) popisující změny

⁵Receiver Operating Characteristic

úspěšnosti klasifikace v závislosti na změně parametrů prahování odezvy modelu. Tento graf byl vytvořen za účelem stanovení prahu pro detekci abnormálního chování objektů.



Obrázek 7.4: ROC křivka – filtrované data

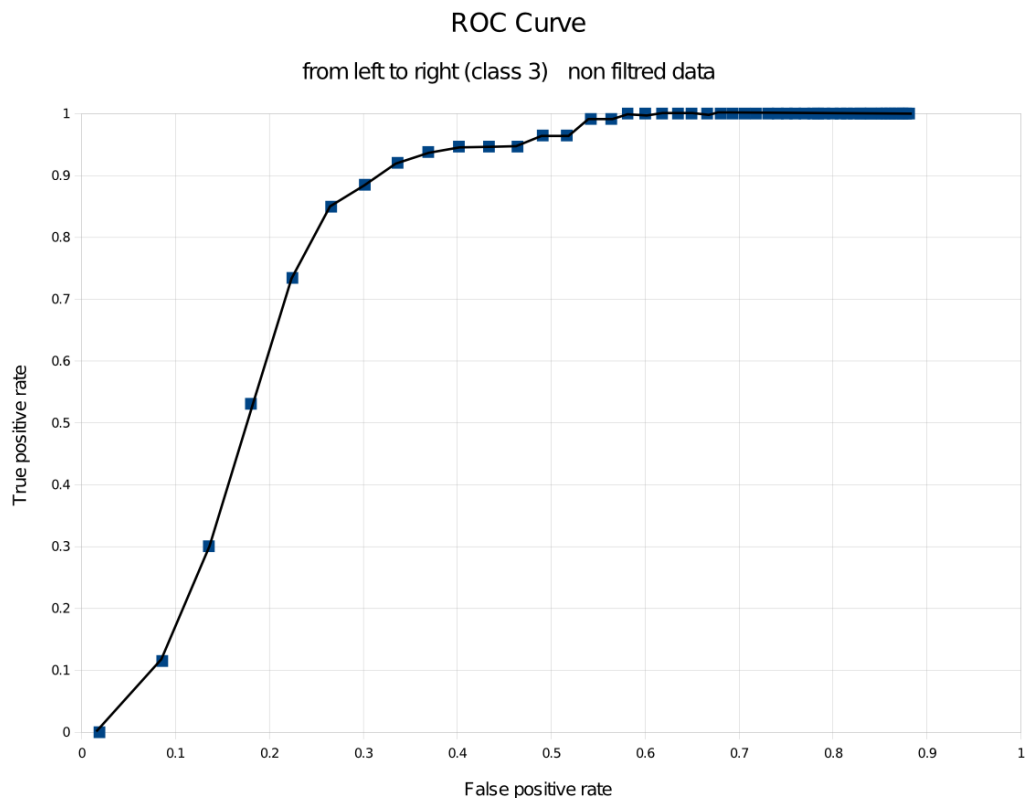
Druhý experiment byl proveden nad nefiltrovanými daty stejným způsobem jako první experiment. Šum byl modelovaný jako samostatná třída (třída č. 4). Přesnost klasifikace na těchto datech byla přibližně 37.63 %. Pro vyhodnocení bylo použito přibližně 10 000 trajektorií. Detailnější výsledky jsou shrnuty v tabulce 7.3.

class	TP [%]	TN [%]	FP [%]	FN [%]
1	1.91	74.54	22.97	0.59
2	0.20	91.02	8.74	0.04
3	0.94	90.71	8.19	0.15
4	34.37	4.06	0.10	61.47

Tabulka 7.3: Výsledky klasifikace nefiltrovaných trajektorií

Pro nefiltrované data byla pro třídu č. 3 sestrojena ROC křivka (viz. obrázek 7.5). Porovnáním obou křivek lze částečně zhodnotit kvalitu algoritmu pro sledování objektu a míru překrytí chování modelovaných pomocí HMM.

Reprezentace výsledku klasifikace trajektorií je demonstrována na obrázku 7.6. Na tomto obrázku stojí před turniketem slečna v růžovém tričku, kolem ní je zobrazený obdélník popisující její současnou polohu. Čárou procházející středem obdélníků detekovaných poloh



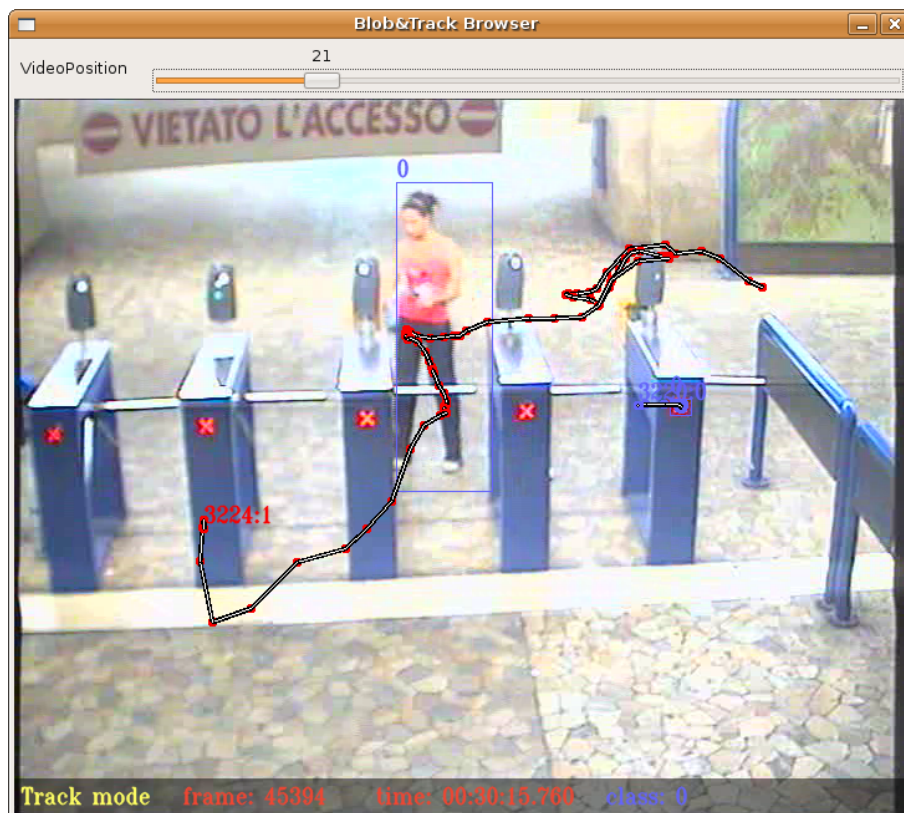
Obrázek 7.5: ROC křivka – nefiltrované trajektorie

je znázorněna její předcházející i následující trajektorie. V pravé části trajektorie se čára kříží, což naznačuje, že trajektorie je ovlivněna více překrývajícími se chováními. Na levé straně trajektorie je vidět systematická chyba algoritmu pro sledování objektů, která by mohla způsobovat problémy při rozpoznání netypického chování. Tato systematická chyba algoritmu pro sledování objektů spočívala v chybné predikci polohy objektu, který se už nenacházel ve scéně. Při modifikaci algoritmu pro sledování objektů je nutné provést novou anotaci dat a přetrénování modelů, tak aby se HMM bylo schopno vypořádat se systematickými sledování objektů.

Možnosti dalšího vývoje

Existuje velké množství možných modifikací detektoru abnormálního chování založeného na trajektoriích, které by pravděpodobně zlepšily výslednou přesnost klasifikace a detekce abnormálního chování.

Jedním z hlavních problémů současného přístupu je nutnost ručně označit velké množství trajektorií a následně provést trénování pro každou scénu, která se má zpracovávat. Pro omezení množství anotací lze provést shlukovou analýzu nad trajektoriemi, které neobsahují abnormální chování a pro jednotlivé shluky použít pro trénování HMM. Problém shlukování trajektorií je proti shlukování jiného typu dat odlišný v tom, že trajektorie mají variabilní délku. Jednou z možností jak obejít tento problém je pomocí umělého vytvoření trénovací sady. Další možností je natrénování modelu na všech získaných datech a jeho následné rozdělení na dva modely v místě největšího rozptylu.



Obrázek 7.6: Ukázka klasifikace trajektorií

Pro zobecnění algoritmu na více scén je nutné provést přepočítání souřadnic do 3D prostoru a analyzovat rozdíly statických parametrů jednotlivých scén. Pro zlepšení účinnosti trénování a přesnosti klasifikace se nabízí možnost použití diskriminativního trénování modelů [7]. Zdá se, že nejvíce limitujícím faktorem pro správnou a kvalitní detekci abnormálního chování je nepřesnost vstupních dat (v podobě trajektorií).

Dalším poměrně zajímavým problémem je nalezení metody, kterou by bylo možno popsat a dobře detekovat překrývající se chování, případně rozdělení trajektorie na dvě samostatné, z nichž by každá reprezentovala jednoznačně jedno chování.

Kapitola 8

Závěr

Cílem této diplomové práce bylo prostudovat metody sledování objektů a implementovat je s ohledem na využití pro sledování objektů v hromadné dopravě a zabudování této implementace do širšího programového systému. Tento úkol se podařilo splnit.

V rámci prozkoumání současných přístupů k řešení tohoto problému jsou v kapitole 2 popsány možnosti dekompozice problému a kapitolách 3 až 5 řešení dílčích kroků potřebných k vyřešení problému.

Konkrétní návrh implementace algoritmů pro vyhledání objektů je řešen v kapitole 7. Výsledkem této snahy je návrh jednoduchého rozhraní na datové úrovni v podobě XML schématu popsaného pomocí Relax NG a rozhraní na úrovni API C++, které jsou uvedeny v příloze A. Pro usnadnění integrace do programového systému byla struktura programu popsána pomocí ER diagramu, který je uveden v příloze B.

Implementace algoritmu pro sledování objektu a algoritmy implementující řešení dílčích problémů jsou na příloženém CD. Kapitola 6 se zabývá přizpůsobením systému pro aplikaci v hromadné dopravě. V rámci projektu Caretaker byly jako rozšíření implementovány algoritmy posouvající problém sledování objektů ve videosekvenci i na analýzu těchto dat. Dosažené výsledky a nedostatky algoritmu, z nichž hlavní jsou složitost nastavení počátečních parametrů systému a nízká přesnost výsledků, jsou blíže diskutovány v kapitole 7. V této kapitole jsou rovněž shrnuty možnosti dalšího pokračování práce, jako jsou například shluková analýza sekvenčních dat, zlepšování kvality výstupu metod pro sledování objektů pomocí kombinace různých metod pro analýzu objektu nebo automatického nastavování metody pomocí měření přesnosti detekce.

Literatura

- [1] Abdelkader, M. F.; Chellappa, R.; Zheng, Q.; aj.: Integrated Motion Detection and Tracking for Visual Surveillance. In *ICVS '06: Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*, Washington, DC, USA: IEEE Computer Society, 2006, ISBN 0-7695-2506-7, str. 28.
- [2] Beran, V.: Matematická morfologie. http://www.fit.vutbr.cz/study/courses/ZPO/private/lectures/zpo_08_mat_morfologie.pdf, 2006.
- [3] Beran, V.: Nástroj pro anotaci a prohlížení dat. 2008.
- [4] Beran, V.; Herout, A.; Hradiš, M.; aj.: Video Summarization at Brno University of Technology. In *ACM Multimedia*, Association for Computing Machinery, 2007, ISBN 978-1-59593-702-5, str. 4.
- [5] Bishop, C. M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006, ISBN 0387310738.
- [6] Bourke, P.: Lens Correction and Distortion. <http://local.wasp.uwa.edu.au/~pbourke/projection/lenscorrection/>, 2002.
- [7] Burget, L.; Matějka, P.; Černocký, J.: Discriminative Training Techniques for Acoustic Language Identification. In *Proceedings of ICASSP 2006*, 2006, s. 209–212.
- [8] Comanciu, D.; Meer, P.: Mean Shift: A Robust Approach, Toward Feature Space Analysis. 2002.
- [9] Comaniciu, D.; Ramesh, V.; Meer, P.: Kernel-Based Object Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, ročník 25, č. 5, 2003: s. 564–575.
- [10] Figueroa, P.; Leite, N.; Barros, R. M. L.; aj.: Tracking Soccer Players using the Graph Representation. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, Washington, DC, USA: IEEE Computer Society, 2004, ISBN 0-7695-2128-2, s. 787–790.
- [11] Forsyth, D. A.; Ponce, J.: *Computer Vision: A Modern Approach*. Prentice Hall, August 2002, ISBN 0130851981.
- [12] Han, J.; Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000, ISBN 1-55860-489-8.
- [13] Intel: *OpenCV (Open Source Computer Vision) Library*. 2007. URL <http://opencvlibrary.sourceforge.net/>

- [14] International Organization for Standardization: ISO/IEC JTC1/SC29 WG11 – Coding of audio, picture, multimedia and hypermedia information. <http://isotc.iso.org/livelink/livelink/fetch/2000/2122/327993/customview.html?func=ll&objId=327993>, 2007, cit. 12.5.2008.
- [15] Isard, M.: *Visual Motion Analysis by Probabilistic Propagation of Conditional Density*. Dizertační práce, Oxford University, 1998.
- [16] Jahne, B.; Geissler, P.; Haussecker, H. (editoři): *Handbook of Computer Vision and Applications*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, ISBN 0123797705.
- [17] Jelínek, T.: *Detekce pohybujících se objektů ve video sekvenci*. Diplomová práce, Brno University of Technology, Faculty of Information Technology, 2007.
- [18] Kang, H.; Kim, D.; Bang, S. Y.: Real-Time Multiple People Tracking Using Competitive Condensation. In *ICPR '02: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02) Volume 1*, Washington, DC, USA: IEEE Computer Society, 2002, ISBN 0-7695-1695-X, str. 10413.
- [19] LG Electronics: HMMD Color Space : Perceptual color space suitable for content-based retrieval. <http://www.lg-elite.com/MIGR/cmip/hmmd/hmmd.html>, cit. 05.05.2008.
- [20] Lyons, R. G.: *Understanding Digital Signal Processing (2nd Edition)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004, ISBN 0131089897.
- [21] Odobez, J.-M.: First version of single/multiple object tracking (public version). Technická zpráva, Information Society Technologies, 2006.
- [22] Orwell, G.: *1984*. Praha : Naše vojsko, 1991, 268 s., z angl. orig. přel. Eva Šimečková; doslov Milan Šimečka.
- [23] Pashley, H.; Hays, J. H.; Liu, Y.: Motion Texture. Technická Zpráva CMU-RI-TR-05-61, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, September 2005.
- [24] Potůček, I.: Person Tracking Using Omnidirectional View. In *Proceedings of the 9th conference STUDENT EEICT 2003*, 9, Brno University of Technology, 2003, ISBN 80-214-2379, ISSN 0572-3043, s. 603–607.
- [25] Potůček, I.: Automatic Image Stabilization for Omni-Directional Systems. In *Proceedings of the Fifth IASTED International Conference on VISUALIZATION, IMAGING, AND IMAGE PROCESSING*, ACTA Press, 2005, s. 338–342.
- [26] Potůček, I.: Detekce a parametrizace objektů v obraze, transformace, RANSAC. https://www.fit.vutbr.cz/study/courses/POV/private/lectures/pov_09_detekce_a_parametrizace_objektu_v_obraze_geometricke_transformace_ransac.pdf, 11 2007.

- [27] Ravera, B.; et al.: Deliverable D44: Audio/video event recognition and single/multiple object tracking. Technická zpráva, Kingston University, 2007. URL <http://www.ist-caretaker.org/>
- [28] Roweis, S.; Ghahramani, Z.: A Unifying Review of Linear Gaussian Models. Technická zpráva, University of Toronto, Department of computer science, 6 King's College Road, Toronto M5S 3H5, Canada, 1997.
- [29] Skoupil, D.: Úvod do paradigmat programování. Technická zpráva, Katedra matematické informatiky, Olomouc, 1994, tR-CS-94-01.
- [30] Sonka, M.; Hlavac, V.; Boyle, R.: *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007, ISBN 049508252X.
- [31] Starner, T.: *Visual recognition of American Sign Language using hidden Markov models*. Diplomová práce, Massachusetts Institute of Technology, Media Laboratory, 1995.
- [32] Vacha, P.; Haindl, M.: Image retrieval measures based on illumination invariant textural MRF features. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-733-9, s. 448–454.
- [33] W3C XML Protocol Working Group: SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). Technická zpráva, World Wide Web Consortium, April 2007.
- [34] Webové stránky: BeeTagg Barcode Reader. <http://www.beetagg.com/>.
- [35] Webové stránky: HMM Toolkit STK from Speech@FIT. <http://www.fit.vutbr.cz/research/groups/speech/index.php?id=stk>.
- [36] Webové stránky: HTK Speech Recognition Toolkit. <http://htk.eng.cam.ac.uk/>.
- [37] Webové stránky: Relax NG. <http://relaxng.org/>.
- [38] Webové stránky: TREC Video Retrieval Evaluation. <http://www-nlpir.nist.gov/projects/trecvid/>.
- [39] Webové stránky: Wikipedia. <http://en.wikipedia.org/>,
Particle Filter – http://en.wikipedia.org/wiki/Particle_filter
Blob detection – http://en.wikipedia.org/wiki/Blob_detection
Morfologické operace – http://en.wikipedia.org/wiki/Morphological_image_processing
Normal distribution – http://en.wikipedia.org/wiki/Normal_distribution
Kalmanův filtr – http://en.wikipedia.org/wiki/Kalman_filter
Proj. transf. – http://en.wikipedia.org/wiki/Projective_transformations
Barvený model RGB – <http://en.wikipedia.org/wiki/RGB>
Barvený model CMYK – <http://en.wikipedia.org/wiki/CMYK>
Barvený model HSL and HSV – http://en.wikipedia.org/wiki/HSL_and_HSV
Barvený model YUV – <http://en.wikipedia.org/wiki/YUV>
Barvený model YCbCr – <http://en.wikipedia.org/wiki/YCbCr>

- Receiver operating characteristic – http://en.wikipedia.org/wiki/ROC_curve
 Gamut – <http://cs.wikipedia.org/wiki/Gamut>
 Sobel operator – http://en.wikipedia.org/wiki/Sobel_operator
 MPEG – http://en.wikipedia.org/wiki/Moving_Picture_Experts_Group
 MPEG-7 – <http://en.wikipedia.org/wiki/MPEG-7>.
- [40] Webové stránky: MDM/KDD2006 Seventh International Workshop on Multimedia Data Mining. <http://www.fortune.binghamton.edu/MDM2006/>, 2006, cit. 2007-07-10.
- [41] Webové stránky: Caretaker. Information Society Technologies, 2008, <http://www.ist-caretaker.org/>.
- [42] Welch, G.; Bishop, G.: An Introduction to the Kalman Filter. <http://www.cs.unc.edu/~welch/kalman/>, 2006.
- [43] Williams, D. B.; Madisetti, V. (editoři): *Digital Signal Processing Handbook*. Boca Raton, FL, USA: CRC Press, Inc., 1997, ISBN 0849385725.
- [44] Winkler, Z.: Lokalizace - Kalmanův filtr, MCL. <http://robotika.cz/guide/localization/en>, 2003.
- [45] Winkler, Z.: Filtrace - jak se vypořádat s nepřesnými daty. <http://robotika.cz/guide/filtering/en>, 2005.
- [46] Woods, E.; Mason, P.; Billinghurst, M.: MagicMouse: an inexpensive 6-degree-of-freedom mouse. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, New York, NY, USA: ACM, 2003, ISBN 1-58113-578-5, s. 285–286.
- [47] Young, S.; Odell, J.; Ollason, D.; aj.: *The HTK Book*. Machine Intelligence Laboratory, Cambridge University Engineering Department, 2006.
- [48] Zemčík, P.; Španěl, M.: *Studijní opora k předmětu Zpracování Obrazu*. Brno University Of Technology, Faculty of Information Technology, 2006, 57 s.
- [49] Černocký, J.: Hidden Markov Models – an Introduction. Technická zpráva, Brno University Of Technology, Faculty of Information Technology, 2003.
- [50] Šlapal, J.: Grafy, Nejkratší cesta v grafu. <http://www.math.fme.vutbr.cz/default.aspx?catalog=3&catsrtext=11&catsrfield=38>, 2007, cit. 2008-05-12.
- [51] Španěl, M.: *Rozpoznávání gest ve video sekvencích*. Diplomová práce, Brno University of Technology, Faculty of Information Technology, 2003.
- [52] Španěl, M.: Analýza a extrakce příznaků z textur. https://www.fit.vutbr.cz/study/courses/POV/private/lectures/pov_05_textury.pdf, 2007.
- [53] Španěl, M.: Clustering, shlukování, statistické metody. https://www.fit.vutbr.cz/study/courses/POV/private/lectures/pov_02_statisticke_rozpoznavani.pdf, 10 2007.
- [54] Žára, J.; Felkel, P.; Beneš, B.; aj.: *Moderní počítačová grafika*. Computer Press, 2005, ISBN 8025104540, 628 s.

Dodatek A

Rozhraní modulu pro sledování objektů

Kompatibilita zdrojových kódů

```
class Tracker {

public:
    /**
     * Inicializace paměti
     * @param frameSize velikost jednoho snímku videa
     */

    Tracker(frameSize); // Inicializace počátečních hodnot,

    /**
     * Uvolnění paměti
     */

    ~Tracker();

    /**
     * Funkce, která provede vyhledání trajektorie v následujícím snímku
     * @param Frame je struktura obsahující vstupní obraz
     * @return informace o poloze jednotlivých objektů v daném snímku
     */

    std::vector<Moment> getNextFrame(Frame);

};

struct Moment {

    int label; // identifikuje objekt vzhledem k předchozím snímkům
    int pixelCount;
```



```

    int x_min;
    int x_max;
    int x_avg; // (x_min + x_max)/2
    int x_grav; // x-ová souřadnice těžiště
    int y_min;
    int y_max;
    int y_avg;
    int y_grav;
}

```

Schéma XML streamů

Pro definování XML schématu je použito Relax NG. XML dokument odpovídající tomuto schématu popisuje trajektorii objektů ve videosekvenci.

```

element DocumentInformation {
  element source {
    element sourceName { text },
    element sourceURL { text },
    element cameraPosition { text },
    element cameraId { text },
    element sourceNotes { text },
    element start { text },
    element duration { text },
    element notes { text },
  },
  element processedBy {
    element algorithm { text },
    element author { text },
    element startTime { text },
    element endTime { text }?,
    element duration { text }?,
    element notes { text },
  }+
},
element Data {
  element frame {
    element positionFrames { xsd:int },
    element positionMilis { xsd:int },
    element object {
      element id { xsd:int },
      element 2D {
        attribute pixelCount { xsd:int },
        attribute x_min { xsd:int },
        attribute x_max { xsd:int },
        attribute x_avg { xsd:int },

```

```
    attribute x_grav { xsd:int },
    attribute y_min { xsd:int },
    attribute y_max { xsd:int },
    attribute y_avg { xsd:int },
    attribute y_grav { xsd:int }
  }
}*
}*
}
```

Dodatek B

ER Diagram

Entity Relationship Diagram implementované aplikace pro sledování objektů vytvořený pomocí programu *Umberllo UML Modeler*.

